UNIVERSIDAD
DE GRANADA

# FUZZY ASSOCIATION RULES IN BIG DATA

DOCTORAL DISSERTATION
*presented to obtain the*
DOCTOR OF PHILOSOPHY DEGREE
*in the*
INFORMATION AND COMMUNICATION TECHNOLOGY PROGRAM
*by*

**Carlos Jesús Fernández Basso**

Ph.D. Advisors

**María José Martín Bautista & María Dolores Ruiz Jiménez**

DEPARTMENT OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE

Granada, Mayo 2020

Editor: Universidad de Granada. Tesis Doctorales
Autor: Fernández Basso, Carlos Jesús
ISBN: 978-84-1306-634-9
URI: http://hdl.handle.net/10481/63902

This doctoral thesis has been supported by the Spanish Ministries of Science, Innovation and Universities (TIN2017-91223-EXP) and Economy and Competitiveness (TIN2015-64776-C3-1-R); and the Energy IN TIME project from the EU 7º Framework Programme (grant agreement No. 608981) and the COPKIT project from the EU 8 Framework Programme (H2020) research and innovation programme (grant agreement No 786687).

*A mi familia.*

# Agradecimientos

El trabajo de investigación tiende a tener momentos de frustración y dificultad. Es por ello que me gustaría de dedicar estas líneas a las personas que me han acompañado y apoyado durante el camino de mi tesis doctoral. Sin ellos este trabajo no habría sido posible.

En primer lugar, me gustaría dedicar esta tesis a mi familia la cual me ha acompañado y apoyado desde siempre. A mis padres por estar ahí, creyendo en mí y guiándome desde niño. A mi madre por su cariño y su constancia conmigo, y a mi padre por su dedicación, ayuda y por enseñarme a superar los problemas. A mis hermanos porque sin un guitarrista y un pianista en casa no habría sido capaz de abstraerme de mi entorno para encontrar la solución a los problemas en cualquier situación. Este trabajo también va dedicado a mis abuelos: María, José Antonio, Aureliano y María, quienes tienen mucha culpa de que sea tal y como soy gracias a criarme a través de sus enseñanzas, historias y anécdotas. Por último, dedicatoria especial a mi pareja Jesica, quien más ha sufrido las frustraciones y problemas, y ha vivido más de cerca todo este proceso. Ha sido un pilar fundamental al estar siempre a mi lado dispuesta a escuchar dudas, problemas, presentaciones que no le interesaban...Gracias por estar ahí dándome ánimos y fuerza a lo largo de este proyecto.

De esta etapa académica, me gustaría agradecer a mis directoras de tesis, María José y Marilo, por su confianza y por el tiempo que me han dedicado en reuniones, correcciones y explicaciones. Os agradezco enormemente el guiarme y escuchar mis ideas para ayudarme a mejorar como investigador, además de ampliar mi repertorio de comida internacional ;). También me gustaría agradecer a todos mis compañeros doctorandos que me han acompañado en este camino: Alexis, Roberto, Noe, Jero, Pablo, Paco Luque, José Ángel… A Juanga por sus charlas de todo tipo de temáticas, a Víctor por no parar de maquinar, inventar e idear cosas nuevas en cada descanso. A Karel porque ha sido un apoyo y un gran amigo. A Paco Baldán por entretenerme con sus puntos de vista en investigación, política, deporte, etc., así como por estar dispuesto a hablar de cualquier problema. Por último, a Diego porque, desde el momento que elegimos las esquinas del despacho, ha sido un apoyo y ha estado siempre ahí para resolver cualquier duda o acompañarme en cualquier Break.

Por otro lado, también me gustaría agradecer a mis compañeros de grupo de investigación por estar conmigo guiándome y enseñándome. A Juan y Migue, mis compañeros de piso durante unos días en Londres. Por hacer que 3 meses en una cultura extranjera se me pasaran rápido (sin contar la búsqueda de piso), por enseñarme muchísimo, explicarme cantidad dudas, así como guiarme por temáticas que abarcan desde la investigación hasta el descrubrimiento de culturas tan peculiares como la finlandesa. A Jesus Campaña por estar ahí y por animarme a ser extrovertido en los congresos, incluso jugando al futbolín.

*Special thanks to Caren, Ian and Luka for the incredible treatment and hospitality received during the visits to Leicester.*

Por último, agradecer a amigos de toda la vida como Javi por sus charlas y partidos de futbol de desestrese. A Manolo, Javi y Paco, que sin importar el tiempo que pasemos sin vernos, sabemos que siempre estamos ahí. A Jorge, Marta, Beatriz, Raúl y el resto de personas que me han acompañado durante todo este tiempo.

Muchísimas gracias a todos. Este trabajo también es vuestro.

# Table of Contents

# List of Abbreviations

**KDD** Knowledge Discovery in Databases

**ARM** Association Rule Mining

**FAR** Fuzzy Association Rules

**ML** Machine Learning

**MR** MapReduce

**DCS** Distribute Computing using Spark

**OPG** Operational plan generator

**MPC** Model Predictive Control

## Chapter I

# Ph.D. Dissertation

# 1 Introduction

Nowadays data analysis is one of the fastest growing fields in society [WZWD13]. In particular, Big Data analytics aims to obtain knowledge from the behaviour of people and their environment. That is why many companies and governments are betting on these technologies to get more information from their data.

Companies, governments and researchers are constantly looking for interesting information that they have not found before. From now on, this information is the only way to take that next step. All this information comes from a variety of sources, both structured and unstructured. In addition, companies and researchers are dealing with the new challenges of "volume", "speed", "accuracy" and "value" (among many other V's) [SS13] that characterize this new paradigm.

The main concept of Big Data is that this large amount of information will allow Data Mining algorithms to reach better and more accurate models than ever before in an efficient way. Classic Data Mining algorithms are not prepared to work with these new volume and speed requirements.

Data Mining is being one of the hottest issues in Information Technology Research and is one of the most characteristic phases of KDD(Knowledge Discovery in Databases). The KDD process can be defined as the non-trivial process of identifying valid, potentially useful and understandable patterns in the data. The key steps of a KDD process are the following:

1. **Problem specification**: sets the desired target of the discovery and the kind of knowledge the user want to extract.

2. **Data extraction**: analyzes the most important sources of information, and integrates all that sources into a single piece, known as dataset.

3. **Data preprocessing**: reduces, cleans and fixes the data in order to obtain a quality dataset for further stages.

4. **Data mining**: extracts patterns and/or models from the preprocessed data.

5. **Data interpretation and evaluation**: analyzes the extracted knowledge and presents the results in a user friendly shape.

The objective of Data mining stage is to extract non-explicit knowledge that is useful for the user, by constructing a model that is based on the collected data for describing the patterns and relations in the data. Although Data Mining is considered the most important task of the KDD process, it ultimately depends on the data. This requires good processing and techniques to improve the interpretability of these data and their results. The pre-processing of data and the application of fuzzy logic is a crucial step that allows Data Mining algorithms to find more useful patterns with better interpretability.

Data mining covers a wide range of techniques which can be classified into two main types: supervised techniques such as classification methods [Bre01] and non-supervised such as clustering [MBY$^+$16] or association rule mining.

Moreover, attending to the type of the target variable, whether it is defined or not, we can distinguish two different groups:

- **Supervised learning**: the objective is to predict the value of the target variable for new instances by defining the relationship between the input variables and the target variable. They can be grouped into two large groups.
  - *Classification* [DHS12]: the target variable has discrete values, and the different sets of possible outcomes (i.e. classes or labels) are known. For example, the category type of a product.
  - *Regression* [Alp20]: the domain of the target variable is continuous. For example, the forecast of the price of a house.
- **Unsupervised learning**: the target variable is not defined. The objective is to discover the implicit relationships in the data. They can be grouped into two large groups.
  - *Clustering* [Har75]: the objetive is to create groups of similar instances (intra-cluster distance), with as much separation between the groups as possible (inter-cluster distance). For example, grouping groups of customers.
  - *Association* [AIS93b]: discovers hidden patterns of frequent relationships between different characteristics of the data among variables. For instance, the relationships between the purchases of a shopping basket.

Association rule mining (ARM) is based on two phases, the extraction of frequent itemset by means of algorithms such as Apriori [AS+94], Eclat [Zak00] or FP-Growth [HPY00] and the later extraction of the association rules. Additionally, frequent itemsets mining is an important phase in ARM and can also be employed for discovering other types of patterns such as sequential patterns [PHMA+04], gradual dependencies [HÖ2] or exception and anomalous rules [DRS11] to discover meaningful and different patterns amongst them.

Association rules were formally first defined by Agrawal et al. [AS+94] as follows. Let $I = \{i_1, i_2, \dots, i_n\}$ a set of items and $D = \{t_1, t_2, \dots, t_N\}$ a set of $N$ contains transactions each of which a subset of items. In this ambient an association rule can be defined as follows:

$$X \to Y, \text{ where } X, Y \subseteq I \text{ and } X \cap Y = \emptyset. \tag{I.1}$$

$X$ is referred as the antecedent (or left-hand side of the rule) and $Y$ as the consequent (or right-hand side of the rule). The problem of uncovering association rules is usually developed in two steps:

- Step 1: Finding all the itemsets above the minimum support threshold. These itemsets are known as frequent itemsets.
- Step 2: Using the frequent itemsets, association rules are discovered by imposing a minimum threshold for an assessment measure such as confidence.

The most commonly used measures to extract frequent itemsets and association rules are:

- The *support* [BBSV01] is the measure of the frequency with which an item appears in the database. In general, the most interesting association rules are those with a high support value.

$$Supp_D(X) = \frac{|t_i \in D : X \subseteq t_i|}{|D|} \tag{I.2}$$

- Given the itemsets $X$ and $Y$, and the database $D$, the confidence of the rule $X \rightarrow Y$ [BBSV01], represented as $Conf_D(X \rightarrow Y)$, is the conditional probability of $Y$ appearing in those transactions in $D$ that contain $X$.

$$Conf_D(X \rightarrow Y) = \frac{Supp_D(X \cup Y)}{Supp_D(X)} \tag{I.3}$$

## 1.1 Framework and motivation

With the increasing size of the data generated and stored, traditional Data Mining and data preprocessing techniques are facing a great challenge to be able to process large data sets efficiently. For this reason, the use of distributed computing has been used as a solution even before the Big Data phenomenon. This is why it has become mandatory in the Big Data frameworks. This does not only require adapting existing algorithms, but also proposing new ones, to handle Big Data problems.

MapReduce (MR) is one of the first distributed computing paradigms that allowed the generation and processing of Big Data datasets in an automatic and distributed way. MR has become the benchmark in distributed computing paradigms because of its simplicity and fault tolerance. By implementing two functions, *Map* and *Reduce*, users can process large amounts of data without worrying about technical issues such as data partitioning, fault recovery, or job communication.

Apache Hadoop is the most popular and widely used implementation of the MR [Whi12] paradigm. However, despite becoming the benchmark in performance, Apache Hadoop had some limitations such as lack of in-memory processing, or iterative under performance. Thus a novel framework has emerged that focuses on speed and ease of use. This framework, called Apache Spark, solves the limitations of Apache Hadoop by providing memory computing. This is achieved through a novel data structure, called Resilient Distributed Datasets (RDD). The main functions that Spark allows for computer distribution are:

- *Map*: Applies a transformation function to each element of RDD and returns transformed RDD. For instance, a *Map* function applied to $< key, value >$ pairs will give transformed pairs. Figure 1 shows the functioning of this distributed function which is applied to each data partition having the format $< key, value >$ generating as result another pair of the form $< key, transformed\_value >$.

$$Map(< item, value_i >) \rightarrow < item, transformed\_value_i > \tag{I.4}$$

- *Reduce* or *ReduceByKey*: Aggregates the elements of the dataset using an aggregation function. For example, the frequency of appearance of an item is computed by applying a *Reduce* function in the following way:

$$Reduce(< item, list(value) >) \rightarrow < item, value_{aggregated} > \tag{I.5}$$

  As it is shown in Figure 3 the functioning of the *ReduceByKey* function is executed, to add the values with the same key in a distributed way across the cluster.

- *Filter*: This function allows to filter the distributed data according to a condition (see Figure 2).

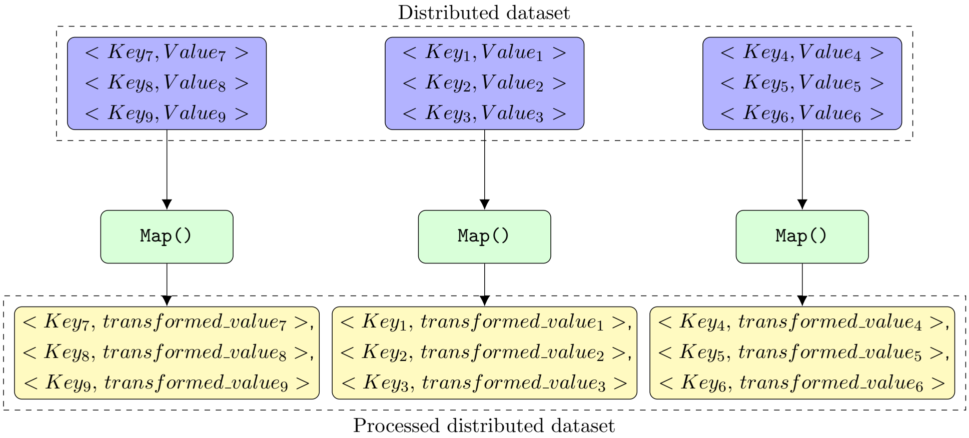

Distributed dataset
$< Key_7, Value_7 >$
$< Key_8, Value_8 >$
$< Key_9, Value_9 >$
$< Key_1, Value_1 >$
$< Key_2, Value_2 >$
$< Key_3, Value_3 >$
$< Key_4, Value_4 >$
$< Key_5, Value_5 >$
$< Key_6, Value_6 >$
Map()
Map()
Map()
$< Key_7, transformed\_value_7 >$,
$< Key_8, transformed\_value_8 >$,
$< Key_9, transformed\_value_9 >$
$< Key_1, transformed\_value_1 >$,
$< Key_2, transformed\_value_2 >$,
$< Key_3, transformed\_value_3 >$
$< Key_4, transformed\_value_4 >$,
$< Key_5, transformed\_value_5 >$,
$< Key_6, transformed\_value_6 >$
Processed distributed dataset

Figure 1: Functioning of Map processing in Spark

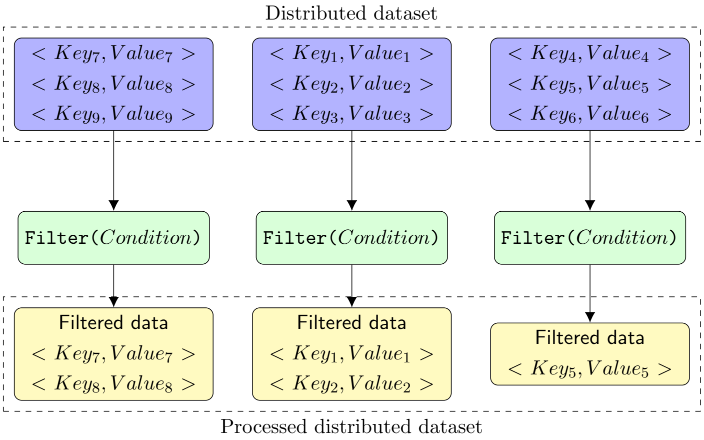


Figure 2: Functioning of Filter processing in Spark

The new Big Data frameworks, such as Apache Spark or Apache Hadoop, include distributed ML libraries. However, only the non-exhaustive methods for extracting frequent itemsets and association rules are included and they do not contain any algorithm for dealing with fuzzy data as well as fuzzy data pre-processing methods. That is why it is necessary to design and develop methods that allow us to extract frequent patterns in large data sets as well as methods that facilitates a better interpretability through fuzzy logic.

The present thesis deals with different topics: mining of association rules and frequent itemsets in Big data with crisp and fuzzy data, extraction of frequent itemsets using Big data techniques in streaming environments, improvement of association rule visualization techniques and application of these algorithms to real data. All topics focus around a common denominator, the Big Data techniques employed for mining interesting information from data. First, a complete study of the most popular Big Data frames to date has been made, in order to draw the current state of the art in terms of power of scarcity, and open and future problems. We have designed different association rules extraction methods using the Big Data framework. Then, we have addressed the use of fuzzy logic in these algorithms, a problem little explored so far. In addition, we have designed

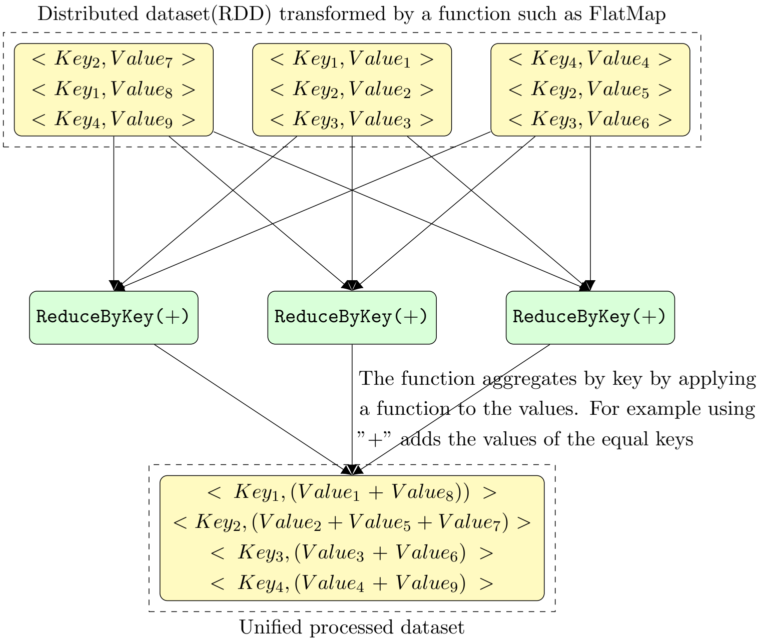

Distributed dataset(RDD) transformed by a function such as FlatMap
$< Key_2, Value_7 >$
$< Key_1, Value_8 >$
$< Key_4, Value_9 >$
$< Key_1, Value_1 >$
$< Key_2, Value_2 >$
$< Key_3, Value_3 >$
$< Key_4, Value_4 >$
$< Key_2, Value_5 >$
$< Key_3, Value_6 >$
ReduceByKey(+)
ReduceByKey(+)
ReduceByKey(+)
The function aggregates by key by applying a function to the values. For example using "+" adds the values of the equal keys
$< Key_1, (Value_1 + Value_8)) >$
$< Key_2, (Value_2 + Value_5 + Value_7) >$
$< Key_3, (Value_3 + Value_6) >$
$< Key_4, (Value_4 + Value_9) >$
Unified processed dataset

Figure 3: Functioning of *ReduceByKey* processing in Spark

an algorithm to extract frequent itemsets in streaming. Finally, we have focused on the visualization and applications of these algorithms to real data sets.

This thesis comprises two different chapters: the Ph.D. dissertation and the publications. In the first chapter, Section 1 provides the general context of this project.The objectives of this thesis are presented in Section 2. Afterwards, the different investigations carried out in this thesis will be explained in Sections 3, 4, 5, 6 and 7, each one with its introduction, development and conclusions. Finally, Section 8 provides the overall conclusions, and open future lines derived from this thesis are in Section 9. The second chapter of the document consists of the different publications that compose this thesis, organized according to the proposed objectives explained before.

# Introducción

Hoy en día el análisis de datos es uno de los campos de más rápido crecimiento en la sociedad. En particular, Big Data Analytics tiene como objetivo mejorar el conocimiento del comportamiento y los procesos de las personas y su entorno. Por ello, muchas empresas y gobiernos apuestan por estas tecnologías para sacar más provecho de sus datos.

Las empresas, los gobiernos y los investigadores están constantemente buscando información interesante que no hayan encontrado antes. Toda esta información proviene de diversas fuentes, tanto estructuradas como no estructuradas. Además, las empresas y los investigadores se enfrentan a los nuevos desafíos de "volumen", "velocidad", "variedad" y "valor" (entre muchas otras V) [SS13] que caracterizan este nuevo paradigma.

El concepto principal de Big Data es que esta gran cantidad de información permitirá que los algoritmos de minería de datos alcancen mejores y más precisos modelos que antes no podrían obtenerse de una manera eficiente. Los algoritmos clásicos de minería de datos no están preparados para trabajar con estos nuevos requisitos de volumen y velocidad por ello, es necesario su desarrollo con técnicas de Big Data.

La mineria de datos es uno de los temas más candentes en la investigación de las tecnologías de la información y además es una de las fases más características del proceso KDD (Knowledge Discovery in Databases). Este proceso puede definirse como el proceso no trivial de identificación de patrones válidos, potencialmente útiles y comprensibles en los datos. El aspecto clave del proceso de KDD son los pasos en los que se divide:

1. **Especificación del problema**: establece el objetivo del análisis y el tipo de conocimiento que se pretende extraer.

2. **Extracción de datos**: analiza las fuentes de información más importantes e integra todas esas fuentes en una única pieza, conocida como conjunto de datos.

3. **Preprocesamiento de datos**: reduce, limpia y arregla los datos con el objetivo de obtener un conjunto de datos de calidad para las etapas posteriores.

4. **Minería de datos**: extrae patrones y/o modelos de los datos preprocesados.

5. **Interpretación y evaluación de los datos**: analiza el conocimiento extraído y presenta los resultados de una forma agradable para el usuario.

El objetivo de esta etapa es extraer conocimiento no explícito que sea útil para el usuario, mediante la construcción de un modelo basado en los datos recogidos y la descripción de las relaciones entre los datos. Aunque la minería de datos se considera la tarea más importante del proceso de KDD, en última instancia depende de los datos. Esto requiere un buen procesamiento y técnicas eficientes para mejorar la interpretabilidad de estos datos y sus resultados. El preprocesamiento de los datos y la aplicación de la lógica difusa es un paso crucial que permite a los algoritmos de minería de datos encontrar patrones y mejorar la interpretabilidad.

La minería de datos abarca una amplia gama de técnicas que pueden clasificarse en dos tipos principalmente: técnicas supervisadas, como los métodos de clasificación [Bre01], y no supervisadas, como el de agrupamiento de datos, también conocido como clustering [MBY$^+$16] o las reglas de asociación.

Además, atendiendo al tipo de la variable objetivo, definida o no, podemos distinguir dos grupos diferentes:

- **Aprendizaje supervisado**: el objetivo es predecir el valor de la variable objetivo para nuevas instancias definiendo la relación entre las variables de entrada y la variable objetivo. Se pueden agrupar en dos grandes grupos:
  - *Clasificación* [DHS12]: la variable objetivo es un valor discreto, y se conocen los diferentes conjuntos de resultados posibles (es decir, clases o etiquetas). Por ejemplo, el tipo de categoría de un producto.
  - *Regresión* [CM07]: el dominio de la variable objetivo es continuo. Por ejemplo, la previsión del precio de una casa.

- **Aprendizaje no supervisado**: la variable objetivo no está definida. El objetivo es descubrir las relaciones implícitas en los datos. Se pueden agrupar en dos grandes grupos:
  - *Clustering* [Har75]: permite crear grupos de instancias similares (distancia intra-grupal), con la mayor separación posible entre los grupos (distancia entre grupos). Por ejemplo, agrupar grupos de clientes.
  - *Asociación* [AIS93b]: descubre patrones ocultos de relaciones frecuentes entre diferentes características del conjunto de datos. Por ejemplo, la búsqueda de relaciones entre las compras de una cesta de la compra.

La minería de reglas de asociación (ARM) se basa en dos fases: la extracción de conjuntos de elementos frecuentes mediante algoritmos como Apriori [AS$^+$94], Eclat [Zak00] o FP-Growth [HPY00] y posteriormente la extracción de las reglas de asociación. Además, la extracción de conjuntos de elementos frecuentes es una fase importante en ARM y también puede emplearse para descubrir otros tipos de patrones como los secuenciales [PHMA$^+$04], las dependencias graduales [HÖ2] o las reglas de excepción y anómalas [DRS11] para descubrir patrones significativos y diferentes entre ellos.

Las reglas de asociación fueron formalmente definidas por primera vez por Agrawal et al. [AS$^+$94] de la siguiente manera. Siendo $I = \{i_1, i_2, \dots, i_n\}$ un conjunto de ítems y $D = \{t_1, t_2, \dots, t_N\}$ un conjunto de $N$ transacciones que contiene un subconjunto de ítems. En este ambiente una regla de asociación puede ser definida como sigue:

$$X \to Y, \text{ donde } X, Y \subseteq I \text{ y } X \cap Y = \emptyset. \tag{I.6}$$

donde a $X$ se le denomina el antecedente (o lado izquierdo de la regla) e $Y$ el consecuente (o lado derecho de la regla). El problema de descubrir las reglas de asociación se desarrolla generalmente en dos pasos:

- Paso 1: Encontrar todos los ítems por encima del umbral mínimo de soporte. Estos conjuntos de ítems se conocen como conjuntos de ítems frecuentes.

- Paso 2: Utilizando los conjuntos de ítems frecuentes, las reglas de asociación se descubren imponiendo un umbral mínimo para una medida de evaluación como la confianza.

Las medidas más utilizadas para extraer conjuntos de ítems frecuentes y reglas de asociación son:

- El *soporte* [BBSV01] es la medida de la frecuencia con que un ítem aparece en la base de datos. En general, las reglas de asociación más interesantes son las que tienen un alto valor de soporte.
$$Sop_D(X) = \frac{|t_i \in D : X \subseteq t_i|}{|D|} \tag{I.7}$$
- Dados los conjuntos de elementos $X$ e $Y$, y la base de datos $D$, la confianza de la regla $X \to Y$ [BBSV01], representada como $Conf_D(X \to Y)$, es la probabilidad condicional de que $Y$ aparezca en aquellas transacciones en $D$ que contengan $X$.
$$Conf_D(X \to Y) = \frac{Sop_D(X \cup Y)}{Sop_D(X)} \tag{I.8}$$

### Marco de trabajo y motivación

Con el aumento del tamaño de los datos generados y almacenados, las técnicas tradicionales de extracción y preprocesamiento de datos se enfrentan al gran reto de poder procesar grandes conjuntos de datos de manera eficiente. Por esta razón, el uso de la computación distribuida se ha utilizado como solución incluso antes del fenómeno de los grandes datos. Por eso se ha convertido en obligatorio en el ambiente Big Data. Esto no solo requiere adaptar los algoritmos existentes, sino también proponer nuevos, nacidos de y para manejar los problemas del tipo Big Data.

MapReduce (MR) es uno de los primeros paradigmas de computación distribuida que permitió la generación y el procesamiento de conjuntos de datos de Big Data de forma automática y distribuida. MR se ha convertido en el punto de referencia en los paradigmas de computación distribuida debido a su simplicidad y tolerancia a los fallos. Al implementar dos funciones, *Map* y *Reduce*, los usuarios pueden procesar grandes cantidades de datos sin preocuparse por cuestiones técnicas como la partición de los datos, la recuperación de fallos o la comunicación del trabajo.

Apache Hadoop es la implementación más popular y ampliamente utilizada del paradigma MR [Whi12]. Sin embargo, a pesar de convertirse en el punto de referencia para mejorar el rendimiento, Apache Hadoop tenía algunas limitaciones como la falta de procesamiento en memoria, o el bajo rendimiento iterativo. Por lo tanto, ha surgido un marco novedoso que se centra en la velocidad y la facilidad de uso. Este herramienta, llamada Apache Spark, resuelve las limitaciones de Apache Hadoop proporcionando computación en memoria. Esto se logra a través de una novedosa estructura de datos, llamada Resilient Distributed Datasets (RDD). Las principales funciones que Spark nos permite para la computacion distribuida son:

- *Map*: Aplica una función de transformación a cada elemento de la RDD y devuelve la RDD transformada. Por ejemplo, una función *Map* aplicada a los pares $< clave, valor >$ dará pares transformados. La Figura 1 muestra el funcionamiento de esta función distribuida que se aplica a cada partición de datos que tiene el formato $< clave, valor >$ generando como resultado otro par de la forma $< clave, valor\_transformado >$
$$Map(< item, valor_i >) \to < item, valor\_transformado_i > \tag{I.9}$$
- *Reduce* o *ReduceByKey*: Agrega los elementos del conjunto de datos mediante una función de agregación. Por ejemplo, la frecuencia de aparición de un elemento se calcula aplicando una función *Reduce* de la siguiente manera:
$$Reduce(< item, list(valor) >) \to < item, valor_{agregado} > \tag{I.10}$$

Como se muestra en la Figura 3 se ejecuta el funcionamiento de la función *ReduceByKey*, para sumar los valores con la misma clave de forma distribuida por el clúster.

- *Filter*: Esta función permite filtrar los datos distribuidos de acuerdo a una condición (ver Figura 2).

Los nuevos framework de Big Data, como Apache Spark o Apache Hadoop, incluyen bibliotecas ML (Machine Learning) distribuidas. Sin embargo, solo se incluye algún método no exhaustivo de extracción de conjuntos de elementos frecuentes, y no tienen ninguno que funcione con datos difusos ni con métodos de preprocesamiento de datos difusos. Por ello es necesario diseñar y desarrollar métodos que nos permitan extraer patrones frecuentes en grandes conjuntos de datos, así como métodos que nos permitan una mejor interpretación a través de la lógica difusa.

La presente tesis aborda diferentes temas: minería de reglas de asociación e itemsets frecuentes en Big data con datos crisp y difusos, extracción de itemsets frecuentes utilizando técnicas Big data en entornos streaming, mejora de las técnicas de visualización de reglas de asociación y aplicación de estos algoritmos a datos reales. Todos los temas giran en torno a un denominador común, las técnicas de Big Data empleadas para extraer información interesante de los datos.. En primer lugar, se ha realizado un estudio completo de los entornos de Big Data más populares hasta la fecha, con el fin de dibujar el estado actual de las técnicas para estudiar su poder de escalabilidad, los problemas abiertos y futuros. Hemos diseñado varios métodos de extracción de reglas de asociación utilizando Spark. También, hemos abordado el uso de la lógica difusa en estos algoritmos, un problema poco explorado hasta ahora. Además hemos diseñado un algoritmo de extracción de itemsets frecuentes en streaming. Finalmente, nos hemos centrado en la visualización y en las aplicaciones de estos algoritmos en conjuntos de datos reales.

La segunda parte del documento consta de diferentes publicaciones que componen esta tesis, organizadas según los objetivos propuestos.

# 2 Objectives

After presenting the introduction and the framework related to this work, we present the main objectives that have driven this thesis.

They include the study and analysis of Big Data, Association rule mining algorithms and data fuzzification process, as well as the development of distributed algorithms for Big Data environments (e.g. Apache Spark).

**The main objective of this thesis is the analysis, design, implementation, evaluation and visualization of scalable association and fuzzy association rule mining algorithms for Big Data, outperforming sequential solutions.**

In the following list, we describe each objective individually:

1. **To provide distributed association rule mining methods capable of analysing massive data**: The development and design of distributed algorithms for the extraction of frequent itemsets and association rules.

2. **To provide distributed fuzzy association rule mining methods for analysing massive data**: Analysis and design of new distributed methods for extracting fuzzy association rules from fuzzy big data volumes.

3. **To provide a distributed methods for mining frequent itemsets in streaming data**: The analysis, design and development of new methods for extracting frequent itemsets in distributed environments using the Big Data framework.

4. **To propose a standardization for association rules representation that improves visualization methods**: The study and comparison of the current methods for the visualization of association rules as well as the compatibility between them.A representation framework will be proposed to improve the compatibility with the tools that can visualize association rules.

5. **Application in energy efficiency**: As a final objective, the techniques of association rules extraction and fuzzification in Big Data environments are tested on sensor data and energy management in non-residential buildings. Additionally a probabilistic method will be developed for improving energy consumption in non-residential buildings.

The first objective is primarily addressed in Section 3, where a distributed processing approach to association rules extraction is presented. The second objective is addressed in Section 4 where a fuzzy association rules mining approach for Big Data environments is described. The third objective is covered in Section 5, which presents our proposal for the extraction of frequent itemsets using Big Data in dataflow environments. The fourth objective is achieved in Section 6 where it is described the analysis, design and development representation framework for association rules algorithms which improves the visualization and interpretability of results. Finally the fifth objective is covered in Section 7 where two applications using real data for improving energy efficiency in non-residential buildings have been presented.

# 3 Association Rules mining in Big Data

From previous sections, we may assert that there is an increasing gap between the storing and processing capabilities of current systems. New tools for processing these data are emerging nowadays. However, we need the appropriate algorithms, capable of dealing with Big Data problems, for enabling or even improving the association rule mining task. There is an increasing need for scalable and distributed algorithm proposals due to the enormous quantity of data that has to be processed nowadays. Association rules have, nevertheless, received little attention within the Big Data environment.

This section describes the development of association rules algorithms applying Big Data technologies. Because of the need to extract association rules in large data sets such as sensors or social networks and the lack of traditional algorithms to process these datasets, it is necessary to design new algorithms.

The papers associated with this part can be found in Sections 1.1 and 1.2 of Chapter II.

## 3.1 Background

Association rules were formally defined for the first time by Agrawal et al. [AIS93a]. The problem consists in discovering implications of the form $A \to B$ where $A, B$ are subsets of items from $I = \{i_1, i_2, \ldots, i_m\}$ fulfilling that $A \cap B = \emptyset$ in a database formed by a set of $n$ transactions $D = \{t_1, t_2, \ldots, t_n\}$ each of them containing subsets of items from $I$. $A$ is usually referred as the antecedent and $B$ as the consequent of the rule.

In the literature we find different association rule extraction algorithms. Among those widely used and known we can enumerate the following: Apriori, Apriori-TIC, ECLAT and FP-growth. The main idea of the Apriori and Apriori-TID algorithms is to reduce the number of itemsets to be analysed by sorting the transactions by item frequency, and removing non-frequent ones in each step [AS+94]. The ECLAT algorithm [Zak00] uses the TD-list structure to improve the calculations and FP-growth uses an FP-tree structure to reduce the number of queries in the dataset [HPY00]. Of the algorithms listed above, there are versions that use distributed processing techniques. Among the proposals using the Spark framework we can enumerate some distributed versions of Apriori [QGYH14, RKK15]. The YAFIM algorithm presented in [QGYH14] and the R-Apriori developed in [RKK15] are Spark versions of Apriori very similar to our proposal. The main difference is the ordering of the MapReduce phase. They make the loop to search k-itemsets inside the distributed process using a hash tree, meanwhile we make the MapReduce for every k-itemset using a hash table. The problem of the implementations in [QGYH14, RKK15] is that it is very difficult to adapt it for the Apriori-TID, since in every step of the loop the YAFIM and R-Apriori algorithms do not know if a k-itemset is frequent or not, information that is used in the TID list.

However, previous implementations for Apriori are not available. The only available implementation within the Spark library is a distributed version of FP-Growth algorithm called PFP (Parallel FP-Growth) [LWZ+08]. PFP Spark implementation is based on a different structure which does not coincide with the traditional FP-Tree, because there do not exist efficient Spark implementations of distributed trees. The PFP algorithm sorts and divides data in several groups and counts itemsets in each group using MapReduce paradigm. This algorithm has different phases:

- Parallel counting of the number of repetitions of each item using MapReduce.
- Grouping Items: Dividing all the items into $k$ groups. The algorithm obtains a list of groups, where each group is identified by a unique groupID.

- MapReduce phase: Per each transaction it extracts the groups containing the items in it. Afterwards, they are reduced by groupID.
- Aggregation of results: It aggregates the results obtained in previous steps giving as a final result the top $k$ frequent itemsets.

This implementation only returns the frequent itemsets of higher level exceeding the minimum support threshold (i.e. if $ABC$ is a frequent itemset, $A, B, C, AB, AC$ and $BC$ are not retrieved). Note that the PFP also depends on a parameter $k$ set at the beginning of the algorithm. This may be an inconvenient, for instance when mining association rules, since itemsets of different granularity are necessary during the extraction process.
Therefore, the aim of this research is to compare the available PFP Spark algorithm and an implementation developed by us of the YAFIM algorithm, with three new Spark implementations of Apriori, Apriori-TID and ECLAT to discover not only frequent itemsets but also association rules. To this purpose, we have proposed and developed distributed versions for Apriori (DApriori), Apriori-TID (DATID) and ECLAT (DECLAT) for frequent itemset mining, and extended them for Association Rule Mining (ARM), phase that all of them will have in common.

## 3.2 Development

In this thesis three new designs and implementations of different association rule extraction algorithms have been developed DApriori, DATID and DECLAT. These are briefly explained below.

### 3.2.1 DApriori: Apriori Big Data approach

Our distributed version of Apriori to extract frequent itemsets using Spark has two main steps. In the first stage of the algorithm we must have the appearing items in each transaction separately in order to be processed by the Map function. To that aim a FlatMap function is employed (see first and second rows of Figure 4). Right after, the Map function is applied returning pairs of the

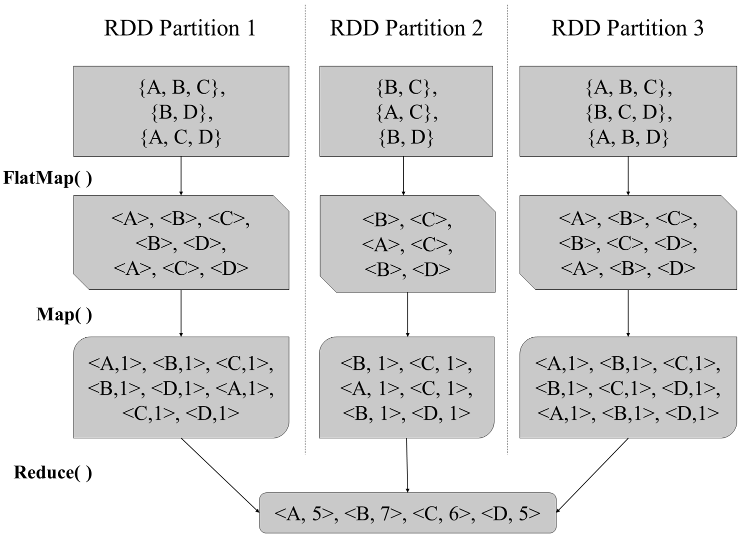

RDD Partition 1
RDD Partition 2
RDD Partition 3
{A, B, C}, {B, D}, {A, C, D}
{B, C}, {A, C}, {B, D}
{A, B, C}, {B, C, D}, {A, B, D}
FlatMap( )
<A>, <B>, <C>, <B>, <D>, <A>, <C>, <D>
<B>, <C>, <A>, <C>, <B>, <D>
<A>, <B>, <C>, <B>, <C>, <D>, <A>, <B>, <D>
Map( )
<A,1>, <B,1>, <C,1>, <B,1>, <D,1>, <A,1>, <C,1>, <D,1>
<B, 1>, <C, 1>, <A, 1>, <C, 1>, <B, 1>, <D, 1>
<A,1>, <B,1>, <C,1>, <B,1>, <C,1>, <D,1>, <A,1>, <B,1>, <D,1>
Reduce( )
<A, 5>, <B, 7>, <C, 6>, <D, 5>

Figure 4: Example of Phase 1 using the example dataset for DApriori and DATID algorithms

form $< item, 1 >$ when *item* has appeared in a transaction (third row of Figure 4). Finally the Reduce phase is in charge of merging items with the same key filtering only those items exceeding the MinSupp threshold.

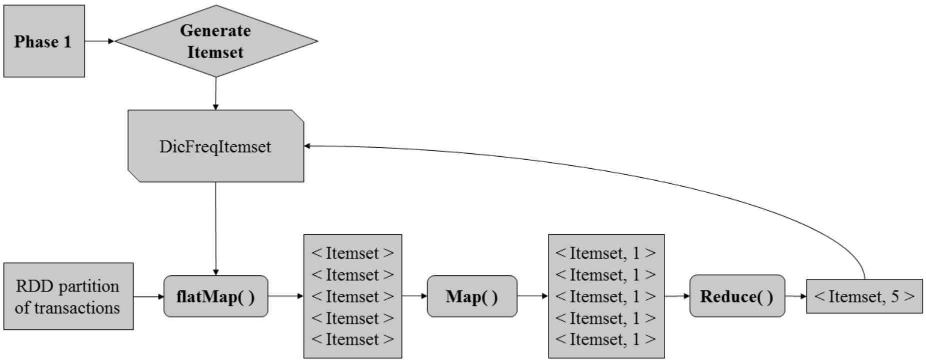


Figure 5: Phase 2 of DApriori algorithm using Spark

During the second phase, the list of frequent items resulting from the previous stage are utilised to form the candidate itemsets of superior length using the downward closure property of frequent itemsets. For the new candidates the same process followed in the first stage is applied, i.e. a FlatMap followed by a MapReduce process plus a filtering step by their support. In Figure 5 we can see the whole process for the second stage of the algorithm. This phase is executed repeatedly until no more candidates of higher length can be found.

### 3.2.2 DATID: Apriori-TID Big Data approach

The Apriori-TID Big Data approach follows a similar structure to DApriori Algorithm. We can find two main differences. The first change before starting the second phase. In this case, the algorithm sorts by support the frequent items obtained in the first stage and then removes the non-frequent items from the set that will be processed in the second phase (see Figure 6). For this reason, the second stage differs from the Apriori, since the computation of frequent itemsets is not performed on the original data. Instead, this computation is made using the transformed data, i.e data which only contain frequent items. Thanks to this modification, the second phase that retrieves itemsets of length 2, and therefore subsequent k+1-itemsets, is faster and decreases memory consumption.

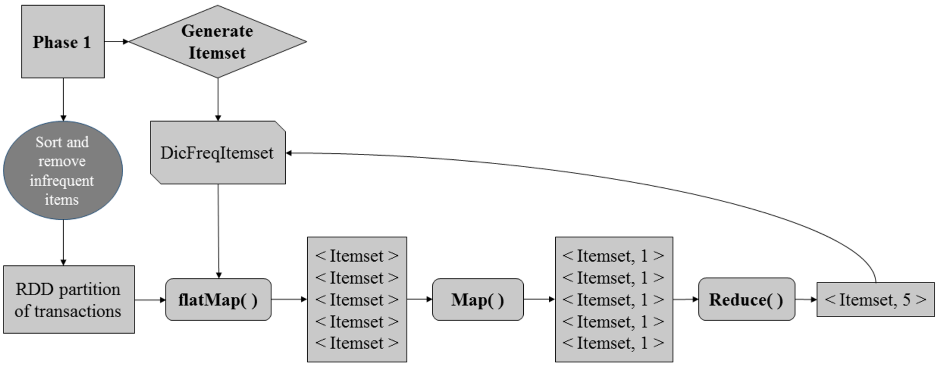


Figure 6: Phase 2 of DATID algorithm using Spark

### 3.2.3 DECLAT: ECLAT Big Data approach

We have also proposed the DECLAT algorithm following the philosophy of the sequential ECLAT algorithm, but using the MapReduce paradigm in the Spark framework. In this case, the principal difference resides in the data distribution by itemsets, instead of distributing the process by set of transactions like in DApriori.
This algorithm has a preprocessing phase, because the database has to be changed to consider items as transactions. In Figure 7 we can see the two main distributive functions for preprocessing the data. First step uses FlatMap, returning for each transaction a pair $< TransactionID, Item >$ if the item appears in the transaction. Secondly the function $GroupByKey$ aggregates the pairs and returns pairs $< Item, [ListTransaction] >$ where the transaction list contains 1 or 0 depending whether the item is in the transaction or not respectively (see an example in Figure 7). DECLAT is also comprised of two phases like previous algorithms. In the first phase, the algorithm counts the number of appearances of every item in every transaction. Firstly, it pre-processes data as explained before to generate a pair comprised of an item and a list of transactions. After that, the reduce function calculates frequent items and generates the itemsets candidates for the next step (see Figure 8).

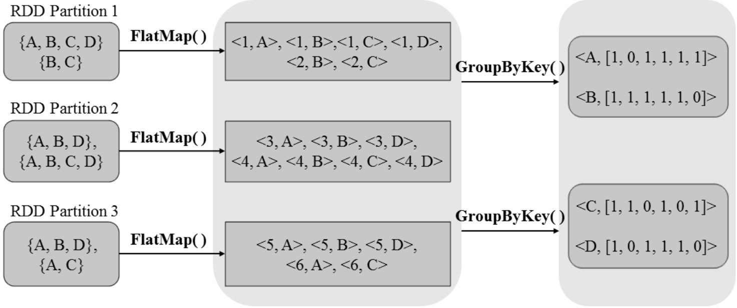


Figure 7: Example of transformation of Data for DECLAT algorithm

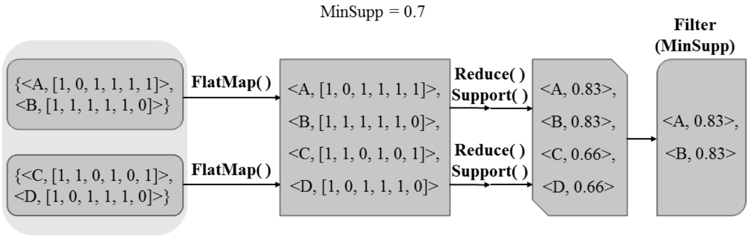


Figure 8: Example of first Phase for DECLAT algorithm

For the computation of the itemsets support, the algorithm employs the $FlatMap()$ function and the itemsets lists like broadcast variables to generate the pair $< Itemset, support >$. Therefore the $FlatMap()$ function returns a pair comprised of an itemset and a list containing 1 or 0 depending if the itemset appear in the transaction or not (see Figure 9). And, finally the *Reduce*() function is applied to obtain the support of the itemsets.

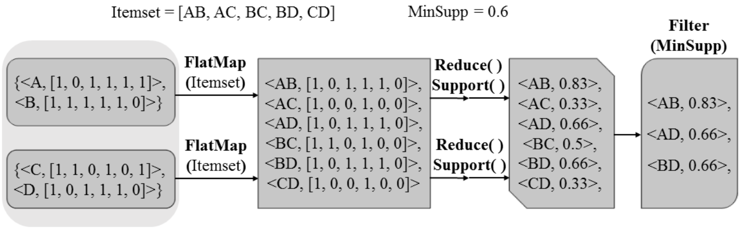


Figure 9: Example of second Phase for DECLAT algorithm

### 3.2.4 Spark Association Rule mining

Once frequent itemsets are extracted, the final step is to uncover the association rules that assess the predetermined thresholds for support and confidence.
The result of previous algorithms is a list of frequent itemsets that will be in RDD format used by Spark. The idea of this algorithm is to use a Map function and afterwords a Reduce function as follows. For each partition the Map function generates the possible rules which are determined by the list of frequent itemsets and it also calculates the confidence of each candidate rule. This will return, for each partition, the list of rules using the following format $< key, value >$ where

- key: represents a rule where the antecedent is separated from the consequent by _ .
- value: contains the confidence of the rule.

To finish, the *Reduce*() function is applied to generate the final set of association rules.

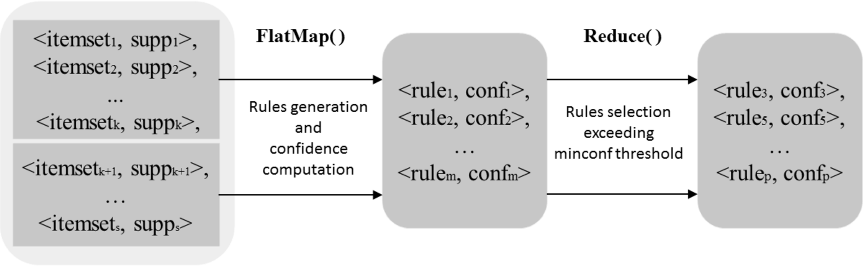


Figure 10: Main procedure for Spark association rule mining

## 3.3 Results and discussion

The results have been obtained by carrying out different experiments with the PFP (available in the Spark library), our implementation of the YAFIM algorithm [QGYH14], DApriori, DATID and DECLAT algorithms in order to compare their performance. Our aim was to study the behaviour of the different available algorithms implemented using Big Data philosophy, in particular using the Spark technology. In order to compare these algorithms, different datasets have been selected to be used in the experimentation to check the behaviour according to the number of items and transactions for the different proposals (see Table I.1).

Table I.1: Dataset and parameter description

| Name | Transactions | Features | Items | MinSupport | MinConfidence |
|---|---|---|---|---|---|
| Poker | 1,025,010 | 11 | 78 | 0.05 | 0.7 |
| Susy | 5 000 000 | 18 | 54 | 0.2 | 0.7 |
| Higgs | 11 000 000 | 28 | 92 | 0.2 | 0.7 |
| Otto | 90000 | 93 | 4300 | 0.2 | 0.7 |

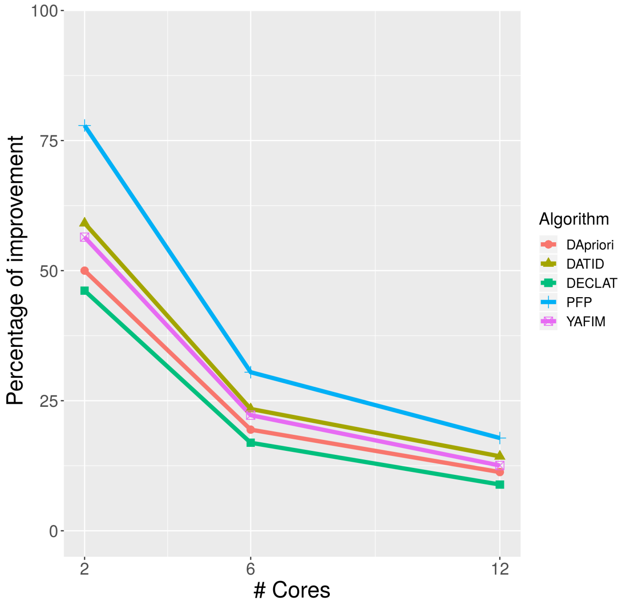


Figure 11: Efficiency of different algorithms for Higgs dataset measured by percentage of improvement

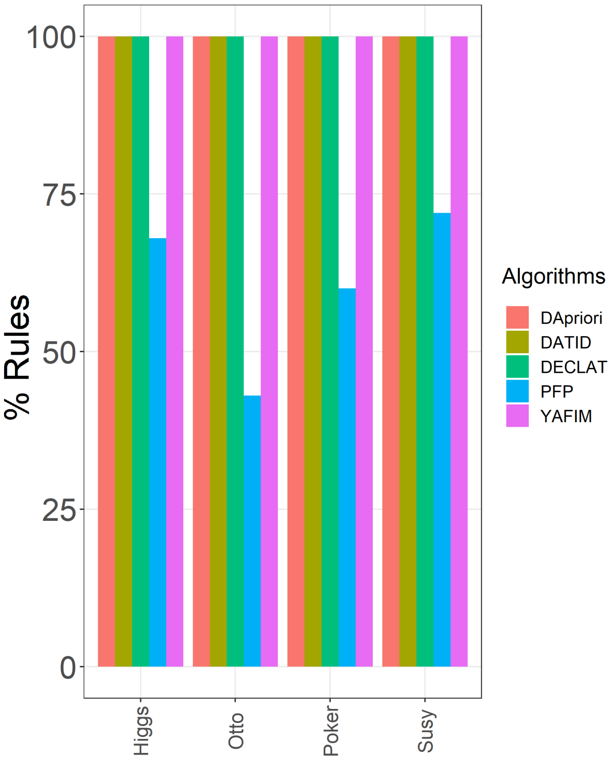


Figure 12: Poker

We have compared these new algorithms with YAFIM algorithm and PFP (FP-Growth algorithm available in the Spark library) obtaining that PFP outperforms DApriori, DATID, DECLAT and YAFIM algorithms as far as the time consumption is concerned (see Figure 11), and DATID improves memory usage. However, the PFP results are not always convenient for their posterior processing to extract association rules, since PFP algorithm only provides the longest frequent itemsets and their support. Therefore when the user is interested in obtaining an exhaustive set of association rules in massive data, the best option available so far is to use DATID algorithm.

After all the performed experiments, it can be observed that the most efficient solution is the Spark PFP. Nevertheless, this algorithm does not provide complete data results (see Figure 12). So depending on the users' needs, a different algorithm should be chosen. For instance, if it is necessary to search the whole set to explore all possible association rules we need to use DATID (Distributive version of Apriori-TID) algorithm. On the other hand, if our search is less deep or we only need to obtain a subset of association rules, e.g. in a recommendation system, the PFP will be faster, although some interesting associations could be missing.

# 4 Fuzzy Association Rules mining in Big Data

In the previous section, we saw that there is a growing need for improving the processing capabilities of current association rules mining algorithms. In addition, there are some proposals that allow a better interpretability and representation of the data that we find in the real world. One of them is the use of fuzzy logic to improve the representation of the data and also improves the interpretability and usefulness of the results obtained.

This section describes the development of fuzzy association rule mining algorithms using Big Data technologies. The papers associated with this part can be found in Section 2.1 and 2.2 of Chapter II.

## 4.1 Background

Association rules were formally defined for the first time by Agrawal et al. [AIS93a]. The problem consists in discovering implications of the form $A \to B$ where $A, B$ are subsets of items from $I = \{i_1, i_2, \dots, i_m\}$ fulfilling that $A \cap B = \emptyset$ in a database formed by a set of $n$ transactions $D = \{t_1, t_2, \dots, t_n\}$ each of them containing subsets of items from $I$. $A$ is usually referred as the antecedent and $B$ as the consequent of the rule.

However, the nature of the data can be diverse and can come described numerically, categorically, imprecisely, etc. In the case of numerical elements, a first approximation could be to categorise them so that, for example, the height of a person may be given by a range to which it belongs, as for instance [1.70, 1.90]. However, depending on how these intervals are defined, the obtained results may vary a lot. To avoid this, the use of linguistic labels such as "high" represented by a fuzzy set is a good option to represent the height of a person having at the same time a meaningful semantic to the user [HY07]. Beside this, we may also have a dataset with inherent imprecise knowledge where ordinary crisp methods cannot be directly applied (see for instance [CDS$^+$04]).

The best to our knowledge, there is only one work presenting how to discover fuzzy association rules employing the MapReduce framework. The approach presented in [GCS16] is based on an extension to the fuzzy case of the Count Distribution algorithm [GICC07, GCC05]. This algorithm uses similar procedure than R-Apriori [RKK15] algorithm where in the second phase, in charge of computing the itemset support, Hadoop is employed instead of Spark. This approach differs from our proposal since we employ the resilient distributed dataset structure that enables across cluster computation. To deal with this kind of data we introduce the concept of fuzzy transaction and fuzzy association rule defined in [BDSV02, DMSV03].

**Definition 1.** *Let $I$ be a set of items. A fuzzy transaction, $t$, is a non-empty fuzzy subset of $I$ in which the membership degree of an item $i \in I$ in $t$ is represented by a number in the range [0, 1] and denoted by $t(i)$.*

By this definition a crisp transaction is a special case of fuzzy transaction. We denote by $\tilde{D}$ a database consisting in a set of fuzzy transactions.

**Definition 2.** *Let $A \subseteq I$ be an itemset, i.e. a subset of items in I. The degree of membership of $A$ in a fuzzy transaction $t \in \tilde{D}$ is defined as the minimum of the membership degree of all its items:*

$$t(A) = \min_{i \in A} t(i). \tag{I.11}$$

**Definition 3.** *Let $A, B \subseteq I$ be itemsets in the fuzzy database $\tilde{D}$. Then, a fuzzy association rule $A \to B$ is satisfied in $\tilde{D}$ if and only if $t(A) \leq t(B)\, \forall t \in \tilde{D}$, that is, the degree of satisfiability of $B$ in $\tilde{D}$ is greater than or equal to the degree of satisfiability of $A$ for all fuzzy transactions $t$ in $\tilde{D}$ .*

Assessment measures for fuzzy association rules have been studied in numerous papers according to different perspectives (see a review in [MRS16]). The approach followed here it is a cardinality based generalization presented in [DMSV03] and extended in other works (see for instance [DRSS11a, RSDMB16]). However, other measures arise in terms of the combination of particular inclusion and cardinality operators. The study made in [DHP06] focuses in studying the suitable operators yielding a fuzzy Ruspini's partition in close relation with negated items in rules. For a deeper discussion on the possible frameworks to assess fuzzy association rules we recommend the review made in [MRS16].

The support and confidence measures are then defined using a semantic approach based on the evaluation of quantified sentences as proposed in [BDSV02] using the $GD$-method [DMSV03] and the quantifier $Q_M(x) = x$, which represents the quantifier *"the majority"*.

**Definition 4.** *The support of a fuzzy itemset $A$ is defined as:*

$$FSupp(A) = \sum_{\alpha_i \in \Lambda} (\alpha_i - \alpha_{i+1}) \frac{|\{t \in \tilde{D} : t(A) \geq \alpha_i\}|}{|\tilde{D}|} \tag{I.12}$$

*where $\Lambda = \{\alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of $\alpha$-cuts.*

**Definition 5.** *The support of a fuzzy association rule $A \to B$ is defined as:*

$$FSupp(A \to B) = \sum_{\alpha_i \in \Lambda} (\alpha_i - \alpha_{i+1}) \frac{|\{t \in \tilde{D} : t(A) \geq \alpha_i \text{ and } t(B) \geq \alpha_i\}|}{|\tilde{D}|} \tag{I.13}$$

*where $\Lambda = \{\alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of $\alpha$-cuts.*

**Definition 6.** *The confidence of a fuzzy association rule $A \to B$ is defined as:*

$$FConf(A \to B) = \sum_{\alpha_i \in \Lambda} (\alpha_i - \alpha_{i+1}) \frac{|\{t \in \tilde{D} : t(A) \geq \alpha_i \text{ and } t(B) \geq \alpha_i\}|}{|\{t \in \tilde{D} : t(A) \geq \alpha_i\}|} \tag{I.14}$$

*where $\Lambda = \{\alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of $\alpha$-cuts.*

Employing support and confidence measures and setting appropriated thresholds for them, fuzzy association rules can be discovered by fixing a set of predefined $\alpha$-cuts [DRSS11a]. Note that considering a sufficiently dense set of $\alpha$-cuts in the unit interval, the obtained measure will be a good approximation of the real measure that should consider every $\alpha \in [0, 1]$ appearing in the dataset. This is the main idea of our proposal for mining fuzzy association rules using MapReduce in Spark framework.

## 4.2 Development

In this section, we present three new algorithms for frequent itemset extraction and the common part of these algorithms for fuzzy association rules (from now on FAR) mining, all of them following the Big Data paradigm for distributed-mode algorithms. These proposals are inspired by the operation of the traditional sequential Apriori, Apriori-TID and ECLAT algorithms and are all implemented using the Spark Framework, which has several facilities for developing Big Data algorithms based on MapReduce and improvements like the use of main memory or advanced DAG. Apache Spark has implemented a data structure to abstract the concept of data partition. This structure is called Resilient Distributed Dataset (RDD) [ZCD$^+$12b]. The RDD concept means that data are distributed across the clusters.

The algorithms that have been designed are three that are composed of 4 phases: the preprocessing phase; phase 1, which processess items; phase 2, which finds k-itemsets; and finally, the association rule mining phase.

### 4.2.1 BDFARE-Apriori

This section is devoted to present the new algorithm, called BDFARE-Apriori (Big Data Fuzzy Association Rules Extraction).

#### Preprocessing

The different algorithms proposed in this thesis for fuzzy association rules mining algorithms pre-process the data by transforming them into an array of BitSets. Other works like [LY00] and [RŠ05], which use this kind of representation, have obtained very good results in terms of memory usage and execution time. The BitSet representation has the advantage of accelerating logical operations such as conjunction or cardinality, which is a fundamental part of our algorithm when calculating item conjunctions and their frequencies.

Therefore, for fuzzy association rule mining, the algorithm processes the data and store them into an array of bit-lists whose size will depend on the number of transactions contained in the chunk, obtaining for each transaction an array of itemsets with their corresponding bit-list. Note that all this process is made in a distributive manner, i.e. for each chunk of data, in order to accelerate the computation. Then, for each transaction and for each item a bit-list will be created containing 1 in position $j$ if the membership value of the item in that transaction is higher or equal than $\alpha_j$ and 0 otherwise. In this way, each item is represented by its bit-list depending on its value in the transaction. For instance, if the itemset $X$ is satisfied with degree 0.25 in a transaction $t_i$, i.e. $\mu_X(t_i) = 0.25$, and the set of $\alpha$-cuts is $\{1, 0.75, 0.5, 0.25\}$, then the associated bit-list of $X$ in that transaction will be $[0, 0, 0, 1]$ (see Figure 13).

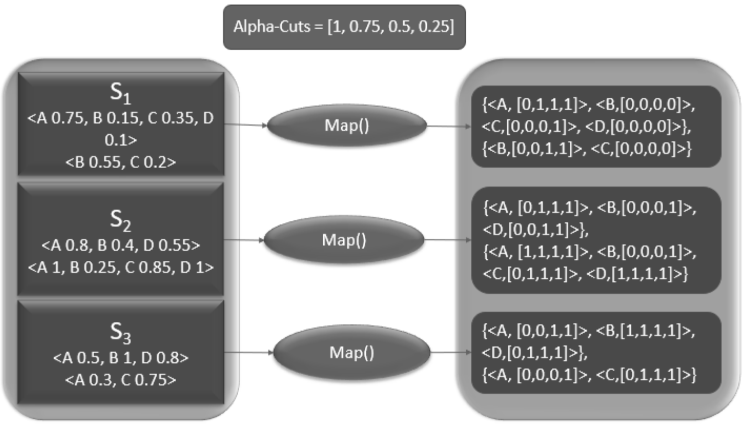

Alpha-Cuts = [1, 0.75, 0.5, 0.25]
S1
<A 0.75, B 0.15, C 0.35, D 0.1>
<B 0.55, C 0.2>
S2
<A 0.8, B 0.4, D 0.55>
<A 1, B 0.25, C 0.85, D 1>
S3
<A 0.5, B 1, D 0.8>
<A 0.3, C 0.75>
Map()
Map()
Map()
{<A, [0,1,1,1]>, <B,[0,0,0,0]>, <C,[0,0,0,1]>, <D,[0,0,0,0]>}, {<B,[0,0,1,1]>, <C,[0,0,0,0]>}
{<A, [0,1,1,1]>, <B,[0,0,0,1]>, <D,[0,0,1,1]>}, {<A, [1,1,1,1]>, <B,[0,0,0,1]>, <C,[0,1,1,1]>, <D,[1,1,1,1]>}
{<A, [0,0,1,1]>, <B,[1,1,1,1]>, <D,[0,1,1,1]>}, {<A, [0,0,0,1]>, <C,[0,1,1,1]>}

Figure 13: Preprocessing data phase which transforms fuzzy items into bit-lists in BDFARE-Apriori

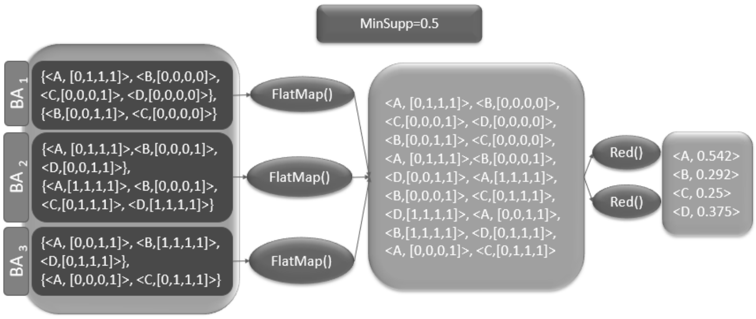

MinSupp=0.5
BA 1
{<A, [0,1,1,1]>, <B,[0,0,0,0]>,
<C,[0,0,0,1]>, <D,[0,0,0,0]>},
{<B,[0,0,1,1]>, <C,[0,0,0,0]>}
BA 2
{<A, [0,1,1,1]>,<B,[0,0,0,1]>,
<D,[0,0,1,1]>},
{<A,[1,1,1,1]>, <B,[0,0,0,1]>,
<C,[0,1,1,1]>, <D,[1,1,1,1]>}
BA 3
{<A, [0,0,1,1]>, <B,[1,1,1,1]>,
<D,[0,1,1,1]>},
{<A, [0,0,0,1]>, <C,[0,1,1,1]>}
FlatMap()
FlatMap()
FlatMap()
<A, [0,1,1,1]>, <B,[0,0,0,0]>,
<C,[0,0,0,1]>, <D,[0,0,0,0]>,
<B,[0,0,1,1]>, <C,[0,0,0,0]>,
<A, [0,1,1,1]>,<B,[0,0,0,1]>,
<D,[0,0,1,1]>, <A,[1,1,1,1]>,
<B,[0,0,0,1]>, <C,[0,1,1,1]>,
<D,[1,1,1,1]>, <A, [0,0,1,1]>,
<B,[1,1,1,1]>, <D,[0,1,1,1]>,
<A, [0,0,0,1]>, <C,[0,1,1,1]>
Red()
Red()
<A, 0.542>
<B, 0.292>
<C, 0.25>
<D, 0.375>

Figure 14: First Phase of BDFARE-Apriori

The implementation of this type of bit-lists have been made using the *numpy* library available in python, which enables concurrent management of lists without going through every list element, enabling the distribution of computations along different clusters. Regarding the use of BitSets, it is necessary to emphasize the low memory resources needed to manage the array of bit-lists created during the preprocessing phase for big amounts of data (see also [DRSS11b, LY00, RŠ05]). In addition, this structure improves the performance of the algorithm since it accelerates conjunction and cardinality computations.

### Phase 1

The first step involves loading the dataset and calculating how often each item appears in the set of transactions using the *Map* and *Reduce* functions. Firstly, we use a *FlatMap* function to transform the lists $[item_1\_bit\_list], \ldots, [item_m\_bit\_list]$ to lists of pairs of the form $< item_i, bit\_list_i >$. We perform this transformation for being able to use a *ReduceByKey*() function that adds all the Bitlists of the same key (i.e. of the same item) to obtain the frequencies of each item. In Figure 14 we can see an example for the value of minimum support threshold of 0.5. In this example, the $FlatMap()$ function, returns the pairs $< item_i, bit\_list_i >$. Afterwards, the items are grouped using the *ReduceByKey*() function using the operator of *numpy* library explained previously, and their support is calculated using the formula in equation (I.12). The last step is in charge of selecting those items with support greater than threshold ($MinSupp = 0.5$). In the example provided in Figure 14 only item $A$ fulfils that condition.

### Phase 2

In the second phase, the algorithm extracts the size-$k$ frequent itemsets. For this task we use different functions: the *CandidateGen*() function generates the itemsets combinations for next $k$ iteration of the algorithm. On the other hand, the function *BitListComputation*() works in a distributed way, the $AND$ combines the candidate itemsets stored in the global variable $Global\_FreqItemset$ through its use in a *FlatMap()* function. The input of this *FlatMap()* is a chunk of the dataset transformed into bit-lists and the variable $Global\_FreqItemset$ containing the frequent itemsets of length $k$. Its output will be a list of pairs comprised of the key of the itemset and its corresponding $bit\_list$ for each transaction. The way to perform this task is to go through all the itemsets in $Global\_FreqItemset$ and search in the transaction each one of the items contained in the itemset. This returns a list with the obtained itemsets and their corresponding bit-list (where the $AND$ operation has been applied to the bit-lists).
Afterwards, these pairs will be grouped by means of a *ReduceByKey*() function which calculates

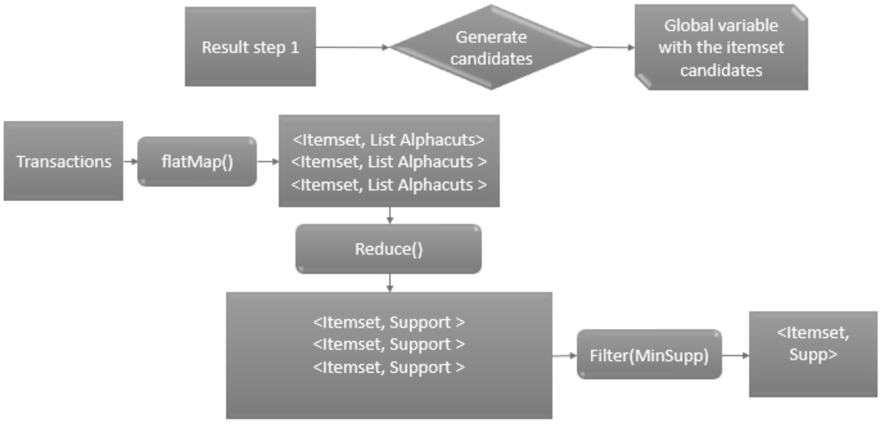


Figure 15: Second Phase of BDFARE-Apriori

the cardinal of the bit-lists. This function will return a list with the items and their cardinal which will be employed to calculate the support of each item.

As mentioned in Section 4.1, we have employed a hash table as the central data structure to accelerate the searching of itemsets. If we use instead of the hash table a linear search at each node, it will result in an increase of time. In [SGM15] a comparison between the different data structures was made concluding that the hash table outperformed among the three data structures (hash tree, trie and hash table) for both real and synthetic datasets.

### Fuzzy Association Rules mining

Once frequent itemsets are extracted, the final step is to uncover the fuzzy association rules that exceed the predetermined thresholds for support and confidence.
The result of previous phases is a list of frequent itemsets that will be in RDD format used by Spark. The idea of this last part of the algorithm is to use a $FlatMap$ function and afterwords a *Reduce* function as follows. For each partition of frequent itemsets the $FlatMap$ function generates the possible rules (see Figure 16). This task is performed by the $GenerateRules()$ function which generates rules from an itemset sequentially [AMS+96]. Therefore the distributed execution of $GenerateRules$ through $FlatMap$ returns pairs of the form $< Rule, Confidence >$, and finally, through a $ReduceByKey$ operation we filter those pairs keeping only those above $MinConf$ threshold.

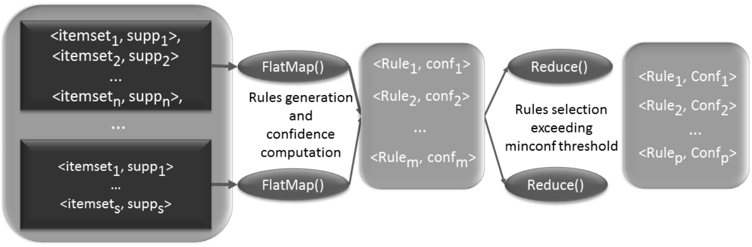


Figure 16: General fuzzy association rules mining procedure using MapReduce

### 4.2.2 BDFARE-Apriori-TID

Another developed algorithm is BDFARE-Apriori-TID, designed following the philosophy of Apriori-TID. In this case, the structure is similar to the BDFARE-Apriori algorithm explained in the previous section.

#### Preprocesing

The preprocessing phase is equal to the BDFARE-Apriori algorithm, because the structure of data for distributive processing across the cluster is the same.

#### Phase 1

The first phase of BDFARE-Apriori-TID algorithm is similar to BDFARE-Apriori. The main difference is that those items with support value lower than the $MinSupp$ threshold are removed from the bit-list vector storage in the *Bit-Array* and sorted after the counting. The ordering followed is from the highest to the lowest according to the number of occurrences of the item in the database.

#### Phase 2

The principal difference with BDFARE-Apriori is that in each iteration of the frequent itemsets searching process all infrequent items are removed.

#### Fuzzy Association Rules mining

The extraction rules phase is equal to the BDFARE-Apriori algorithm explained in Section 4.2.1. We use the hash table obtained with the frequent itemsets for extracting the association rules and calculate the measures like the confidence, lift or certainty factor of each rule.

### 4.2.3 BDFARE-ECLAT

We have also proposed the BDFARE algorithm following the philosophy of the sequential ECLAT algorithm, but using the MapReduce paradigm in the Spark framework. In this case, the principal difference resides in the data distribution by itemset, instead of process distribution by set of transactions like in Apriori.

#### Preprocesing

The preprocessing in BDFARE-ECLAT algorithm is different from the previous explained algorithms. In this way, BDFARE-ECLAT needs the data to be grouped by item joint with their membership degree (in the format of bit-list). Therefore the main difference is the aggregation by item in the MapReduce phase depicted in Figure 17, where there is an example of the aggregation function and output data of BDFARE-ECLAT preprocessing.

Firstly, the algorithm transforms each transaction into pairs $< item_i, bit\_list_i >$ as explained in the preprocesing phase of BDFARE-Apriori algorithm. Then, the algorithm aggregates the different lists by items using a $GroupByKey$ function, generating pairs with an item and a large list of lists containing a bit-list per transaction.

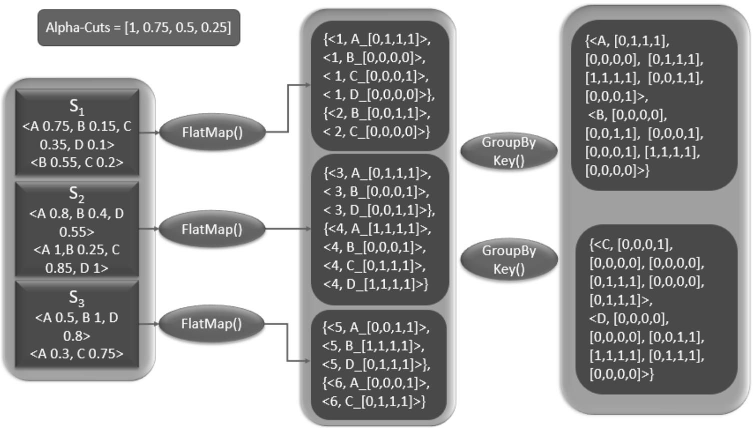

Alpha-Cuts = [1, 0.75, 0.5, 0.25]
S1
<A 0.75, B 0.15, C 0.35, D 0.1>
<B 0.55, C 0.2>
S2
<A 0.8, B 0.4, D 0.55>
<A 1,B 0.25, C 0.85, D 1>
S3
<A 0.5, B 1, D 0.8>
<A 0.3, C 0.75>
FlatMap()
FlatMap()
FlatMap()
{<1, A_[0,1,1,1]>, <1, B_[0,0,0,0]>, < 1, C_[0,0,0,1]>, < 1, D_[0,0,0,0]>}, {<2, B_[0,0,1,1]>, < 2, C_[0,0,0,0]>}
{<3, A_[0,1,1,1]>, < 3, B_[0,0,0,1]>, < 3, D_[0,0,1,1]>}, {<4, A_[1,1,1,1]>, <4, B_[0,0,0,1]>, <4, C_[0,1,1,1]>, <4, D_[1,1,1,1]>}
{<5, A_[0,0,1,1]>, <5, B_[1,1,1,1]>, <5, D_[0,1,1,1]>}, {<6, A_[0,0,0,1]>, <6, C_[0,1,1,1]>}
GroupBy Key()
GroupBy Key()
{<A, [0,1,1,1], [0,0,0,0], [0,1,1,1], [1,1,1,1], [0,0,1,1], [0,0,0,1]>, <B, [0,0,0,0], [0,0,1,1], [0,0,0,1], [0,0,0,1], [1,1,1,1], [0,0,0,0]>}
{<C, [0,0,0,1], [0,0,0,0], [0,0,0,0], [0,1,1,1], [0,0,0,0], [0,1,1,1]>, <D, [0,0,0,0], [0,0,0,0], [0,0,1,1], [1,1,1,1], [0,1,1,1], [0,0,0,0]>}

Figure 17: Preprocessing data phase which transforms fuzzy items into bit-lists in BDFARE-ECLAT

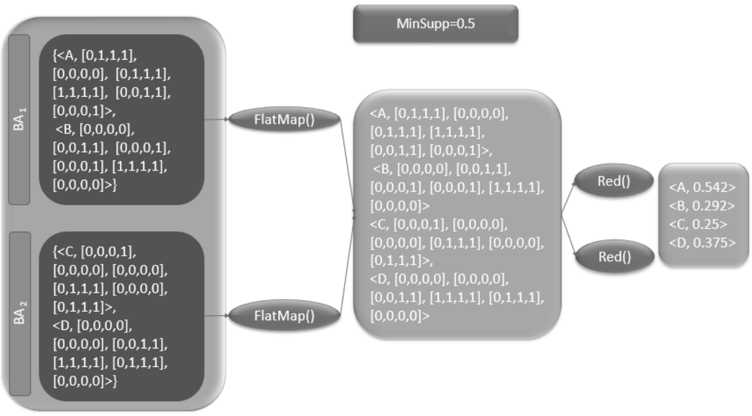

MinSupp=0.5
BA1
BA2
{<A, [0,1,1,1], [0,0,0,0], [0,1,1,1], [1,1,1,1], [0,0,1,1], [0,0,0,1]>, <B, [0,0,0,0], [0,0,1,1], [0,0,0,1], [0,0,0,1], [1,1,1,1], [0,0,0,0]>}
{<C, [0,0,0,1], [0,0,0,0], [0,0,0,0], [0,1,1,1], [0,0,0,0], [0,1,1,1]>, <D, [0,0,0,0], [0,0,0,0], [0,0,1,1], [1,1,1,1], [0,1,1,1], [0,0,0,0]>}
FlatMap()
FlatMap()
<A, [0,1,1,1], [0,0,0,0], [0,1,1,1], [1,1,1,1], [0,0,1,1], [0,0,0,1]>, <B, [0,0,0,0], [0,0,1,1], [0,0,0,1], [0,0,0,1], [1,1,1,1], [0,0,0,0]> <C, [0,0,0,1], [0,0,0,0], [0,0,0,0], [0,1,1,1], [0,0,0,0], [0,1,1,1]>, <D, [0,0,0,0], [0,0,0,0], [0,0,1,1], [1,1,1,1], [0,1,1,1], [0,0,0,0]>
Red()
Red()
<A, 0.542>
<B, 0.292>
<C, 0.25>
<D, 0.375>

Figure 18: First Phase of BDFARE-ECLAT version

### Phase 1

In the first phase, the algorithm counts the number of appearances of every item in every transaction depending on the established set of $\alpha$-cuts. After that, the algorithm calculates frequent items and generates the itemsets for the next step.

The main difference with BDFARE-Apriori resides in how to compute the frequency of items using the list data format. In this case the $FlatMap()$ function (see Figure 18) returns pairs comprised of items and their corresponding *bit-lists* per transaction. Then, the *ReduceByKey*() function calculates the cardinal per item with these data, and afterwards the support is computed.

### Phase 2

For the calculation of itemsets support, the algorithm uses the $FlatMap()$ function in the same way as in the Apriori case. The main difference is how the pair lists are computed. In this case, the $FlatMap()$ function returns a pair comprised of an itemset and a list for each transaction with the *bit-list* using the *numpy* array format (see Figure 19). Therefore, the algorithm calculates all the cardinalities needed for the computation of the cardinality of the obtained itemsets using the *ReduceByKey*() function, and afterwards computes their support. This step will be repeated until no new $k$-itemsets can be found.

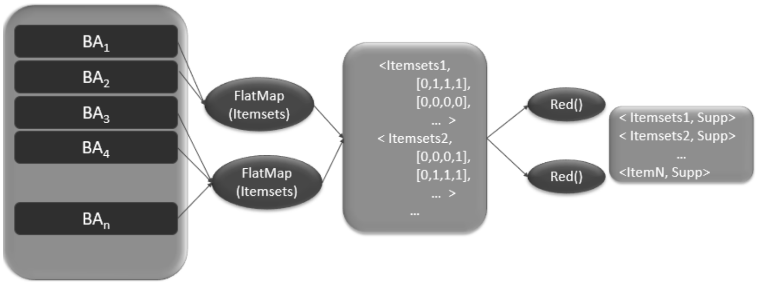


Figure 19: Example of second phase of BDFARE-ECLAT

**Fuzzy Association Rules mining**

The extraction rules phase is equal to BDFARE-Apriori and BDFARE-Apriori-TID previously explained.

## 4.3 Results and discussion

As we have seen, this research has focused on the proposal of fuzzy association rule mining algorithms in the ambit of Big Data, which allows to extract co-occurrence patterns from fuzzy datasets. The traditional algorithms proposed for mining fuzzy association rules can fail when analysing massive datasets due to the memory overflow errors and the very high computational costs, decreasing their efficiency when the dataset grows.
To this end, we have presented and developed new algorithms for mining fuzzy association rules using the Spark framework which enables MapReduce implementations. The different algorithms have been compared and analysed showing improvements not only in the execution time but also in terms of memory, improving the processing capacity, since some of the experiments were not able to end the process in the non-distributed cases. An additional advantage is that our algorithms and their performance can be easily improved even further just by expanding our processing system with more clusters (computation nodes). This allows to scale our approach in external data centers or in cloud systems such as AWS (Amazon Web Services), another great advantage of Big Data technology.

| Fuzzy Database | Transactions | Fuzzy Items |
|---|---:|---:|
| `Bank` | 1446752(45211) | 112 |
| `Higgs` | 11000000 | 86 |
| `Forest-equidepth` | 581012 | 37 |
| `Energy ICPE` | 3649678 | 1121 |

Table I.2: Datasets

The results have been obtained by carrying out different experiments with the different algorithms in order to compare their performance (see Table I.2 for a summary of datasets employed). Figure 20 shows the behaviour of the algorithms when the quantity of resources (1 core means sequential and 24, 48, 72, 102 is equivalent to 1, 2, 3, 4 nodes where each node have 100 Gb of RAM) increases in each of the datasets. It can be observed that the BDFARE-Apriori algorithm performed better than the BDFARE-ECLAT algorithm. Moreover the BDFARE-Apriori-TID is more efficient than the BDFARE-Apriori, as expected.

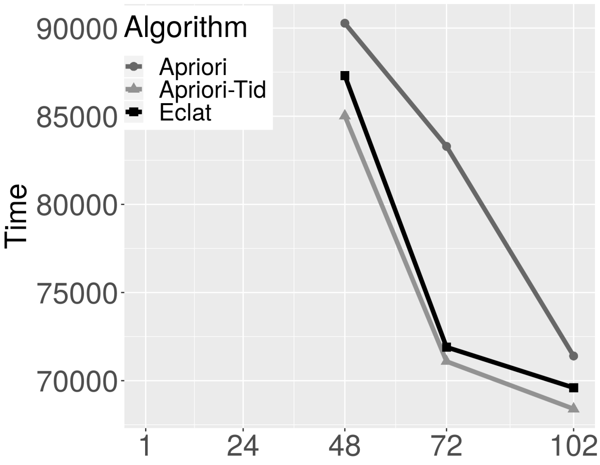


Figure 20: Time for different number of clusters for `Energy ICPE` dataset

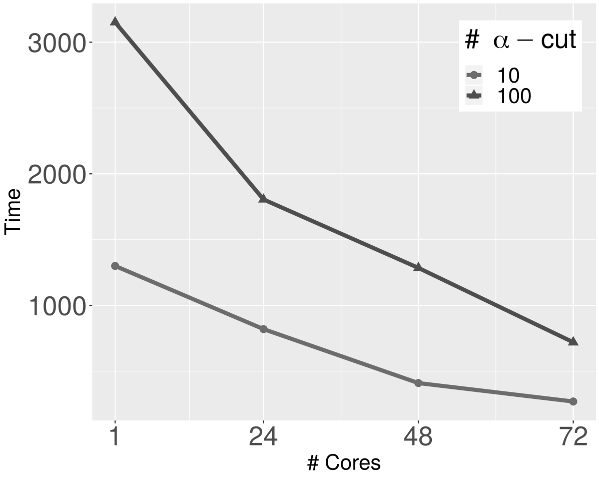


Figure 21: Performance in logarithmic scale of BDFARE-Apriori-TID algorithm with 10 and 100 $\alpha$-cuts when the quantity of cores increases (from 1 core=sequential to 32 cores) in `Bank` dataset

And the BDFARE-ECLAT has some memory problems in `Energy ICPE` dataset, because this dataset has a lot of transactions and items and the lists that BDFARE-ECLAT has to create are very big. For this reason it needs to use a big amount of memory. By contrast, BDFARE-Apriori and BDFARE-Apriori-TID algorithms completed the execution.

In Figures 22 and 23 it can be seen that as the numbers of cores increases, the efficiency and speed up are improved, even they are not optimal. This factor is based on the cores workloads and the network congestion employed for the communication among the cores.

Regarding the performance of BDFARE algorithms with respect to the set of $\alpha$-cuts we have performed different experiments. Figure 21 describes how BDFARE-Apriori-TID algorithm performs when it uses lists to represent $\alpha$-cuts and the number of $\alpha$-cuts is increased, using from 1 to 72 cores.

Additionally, our proposal is based on a decomposition of interestingness measures in terms of $\alpha$-cuts which facilitates their implementation to other interestingness measures different to that of support and confidence using the formal model developed in [DRSS11b], and we have experimentally demonstrated that it is sufficient to consider only 10 equidistributed $\alpha$-cuts in order to mine all significant fuzzy association rules.

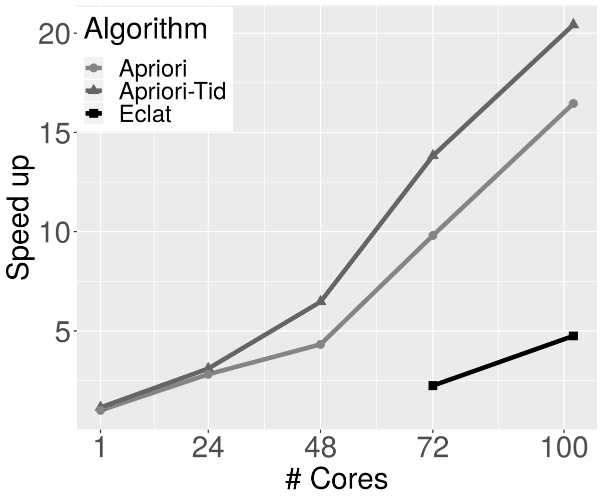


Figure 22: Speed up of BDFARE-Apriori-TID algorithm for Higgs dataset

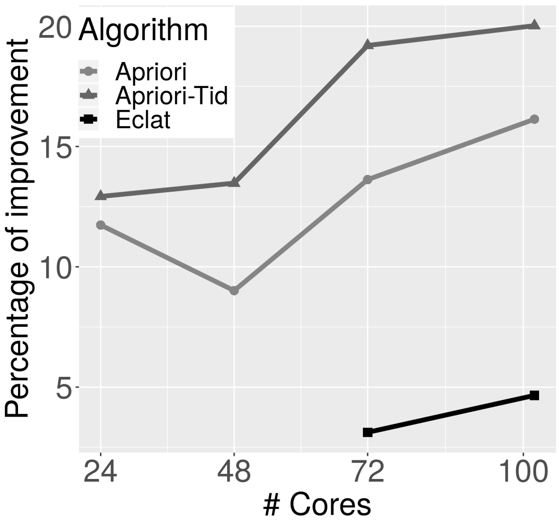


Figure 23: Efficiency of BDFARE-Apriori-TID algorithm for Higgs dataset measured by percentage of improvement

# 5 Big Data frequent itemset mining in streaming

In the previous sections we have studied and developed algorithms to improve the processing of association rules and frequent itemsets using distributed algorithms. But nowadays a lot of data is continuously generated and it is necessary to be able to analyse them in real time.

Recently, the analysis of tendencies is a must due to the necessity of being constantly informed about the economic fluctuations, the social media trends, etc. In this ambit the extraction of frequent items has proved its utility, but the continuous change of data and its enormous volume complicate their extraction with the classic techniques. This section describes the development of a frequent itemsets mining algorithm for streaming environments using Big Data technologies.

The paper associated with this part can be found in Section 3.1 of Chapter II.

## 5.1 Background

In the literature, we can find several approaches to obtain the set of frequent itemsets exceeding a predefined support threshold in sequential and distributive computing, these have been commented on in Section 3.1.

In order to delimit the study of related works we focus on proposals using the sliding window model, because they enable a higher flexibility to define which transactions belongs to the window and it facilitates the windows updating along the time. Among the approaches following this model we distinguish between sliding windows with variable and non-variable length (the amount of transactions in the window does not change).

Among the proposals using *sliding windows with non-variable length*, the CPS-Tree and the GC-Tree algorithms proposed in [TAJL09] and [CL07] resp. store the identified itemsets in a tree structure. The latter employs this structure to extract closed itemsets. On the contrary, in the algorithm developed in [LHL09] the occurrences of each item are stored in a binary vector of bits to accelerate the support updating. The algorithms developed in [LZ11, LZC12] extract maximal itemsets, i.e. all the supersets containing the itemset are infrequent. Despite the time saved by mining maximal itemsets, the main problem is that the entire set of frequent items cannot be recovered from the maximal itemsets.

On the other hand, the approaches using *sliding windows with variable length* are capable to adapt the window in every instant because they enable the computations with a different quantity of transactions. This feature makes them more flexible towards non-periodic systems like transactions from social media. However, non-variable windows are appropriate for sensored ambients where the periodicity of transactions remains almost constant. . The algorithm proposed in [LL09] makes an incremental scanning to update a prefix tree structure where it stores not only the frequency of items, but also some “potential future frequent” items that are infrequent. The algorithm in [KD07] uses also this kind of tree structure and also improves it by applying a greedy approach. However, it presents some drawbacks because its precision is reduced in exchange of memory consumption decreasing. This is due to the fact that items not considered potential items cannot be frequent, although taking into account the transactions in the new coming windows they would be frequent. The proposal developed in [CSXD12] also uses a prefix tree with the itemsets counts and it defines a factor index to express the relevancy of recent transactions. According to this, they make a pruning strategy for the tree which accelerates the memory consumption but decreases the computation efficiency because sometimes the counts of emerging items must be recalculated if they were discarded during the pruning phase.

The FIMoTS (Frequent Itemset Mining over Time-sensitive Streams) algorithm developed in [LZZ$^+$14] has several features that makes it attractive for mining frequent itemsets using variable length sliding windows. We analyse it in more detail paying attention to those aspects of the algorithm that will be used for our proposal using Big Data techniques. One of the optimization techniques that it employs is to classify the itemsets in different categories according to their support, so for every window the membership of the itemsets is revised depending on their frequency in the window. For that aim, the authors define the upper and lower bounds called *Type Transforming Upper/lower Bound*, which are different bounds associated to each itemset to represent the number of transactions that are necessary to change their classification, in particular to be frequent or infrequent. When these bounds reach the zero the support of the itemset is recalculated. Another peculiarity, is that these bounds are expressed as the quotient of two integers in order to facilitate the further computations. The main phases of the FIMoTS algorithm are:

- The *initialisation of the enumeration tree* which contains the supports and the transforming bounds for each item. When the itemset is frequent all its descendants are calculated taking into account the nodes at the same level and when it is not, it is kept but its descendants are not computed.

- The *aggregation and elimination of transactions to the sliding window*. When this happens, the enumeration tree must be recalculated in order to consider the new transactions of the window or the dropped transactions from it. This process consists on the computation of the transforming bounds and in the case that an itemset change its condition from frequent to infrequent or viceversa, the descendants should be prune or recalculated respectively.

## 5.2 Development

Previous revised algorithms cannot be directly applied in distributed clusters. Moreover, the tree structures used in all of them make more complicated their direct extension using the MapReduce framework due to the complexity and low efficiency of operating with distributed trees in different clusters.

The decision of implementing a distributed version of the FIMoTS algorithm is founded on the comparison made in [LZZ$^+$14] where this approach outperforms the reminder approaches using sliding windows. The original FIMoTS algorithm is mainly iterative and performs recurrent updates over the tree structure, but in distributive platforms it is advisable to keep the communication among clusters the lower as possible. However, the maintenance of the tree structure needs direct communication with all the clusters. For these reasons our proposal pays an important attention on this data structure and how the interchange of information is made making it the most efficient as possible. In figure 24 we can see the whole picture of our proposal. In it, we distinguish among the following processes:

- **Initialisation of structures:** The vocabulary (set of different items) is identified and replicated in each slave machine. The transactions coming in the first window are analysed to accomplish the tree initialisation.

- **Data partition:** This is performed using the Spark Streaming which allows the distribution of upcoming transactions till the time interval (window) ends. Once it is closed, we can continue processing the data within it. In the meanwhile the new coming transactions are processed and

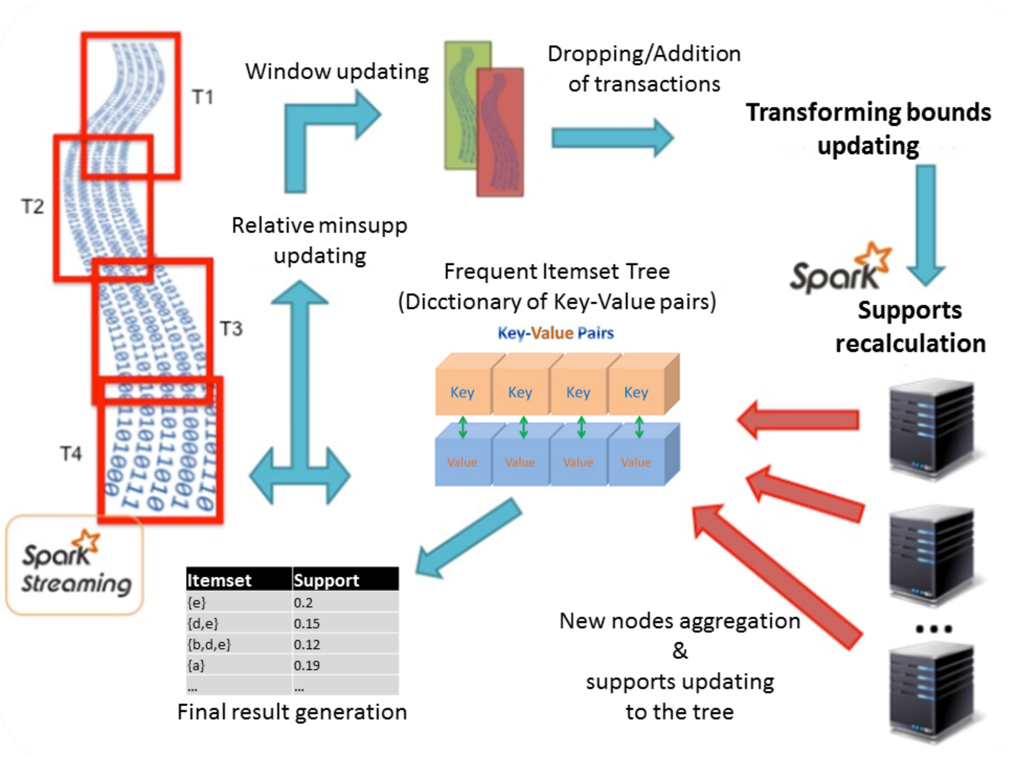

T1
T2
T3
T4
Spark Streaming
Window updating
Relative minsupp updating
Dropping/Addition of transactions
Transforming bounds updating
Spark
Supports recalculation
Frequent Itemset Tree (Dicctionary of Key-Value pairs)
Key-Value Pairs
Key
Value
Itemset
Support
{e}
0.2
{d,e}
0.15
{b,d,e}
0.12
{a}
0.19
...
...
Final result generation
New nodes aggregation & supports updating to the tree
...

Figure 24: Overall schema of FIMoTS

stored in an intermediate state in a RDD (Resilient Distributed Dataset) [ZCD+12a] which is stored in memory for being afterword distributed and processed using the Map and Reduce functions. The main advantage of using RDDs is that we can assure a fault-tolerant system towards fails and delays due to the internal management of RDDs in Spark [ZDLH13].

- **Updates of transforming bounds:** Once the window has moved (some transactions are dropped and new coming transactions are considered), the upper and lower transforming bounds are modified without distributing the process. Since this is a numerical computation process, that does not require a big effort, it is not distributed along the clusters. Hence, it is directly computed over a list where each element is comprised of the upper transforming bound, the lower transforming bound, and all the itemsets with that pair of bounds.

- **Recalculation of supports:** for those itemsets whose transforming bounds have increased/decreased to zero or low, their support is computed considering the transactions in the actual window. This computation is distributed using the MapReduce functions and is recursively applied when the descendants of the itemset must be computed, i.e. when the itemset turns from infrequent to frequent.

- **Frequent itemset tree updating:** Each cluster node returns the obtained information from the distributed process in order to update the support of every itemset. This information is also used to expand or prune the frequent itemset tree. The tree structure is modelled in a global variable which is a dictionary consisting on key-value pairs, where the key is an identifier for the itemset and the value is comprised of all the information about the itemset: its relative support, the quotient of transactions used for the computation of the relative support (numerator, denominator), the identifier of its father node, how many children nodes it has, and a stamp indicating its last modification. This representation of the tree enables a very rapid access from the clusters by indicating the key of the itemset, and also the removal of itemsets when the tree has to be pruned.

## 5.3 Results and discussion

The experiments carried out have been designed to analyse two different aspects:

- To compare our proposal with the non-distributive version of FIMoTS taking into account the execution time and the memory consumption.
- To study the behaviour of our proposal towards different configurations and datasets.

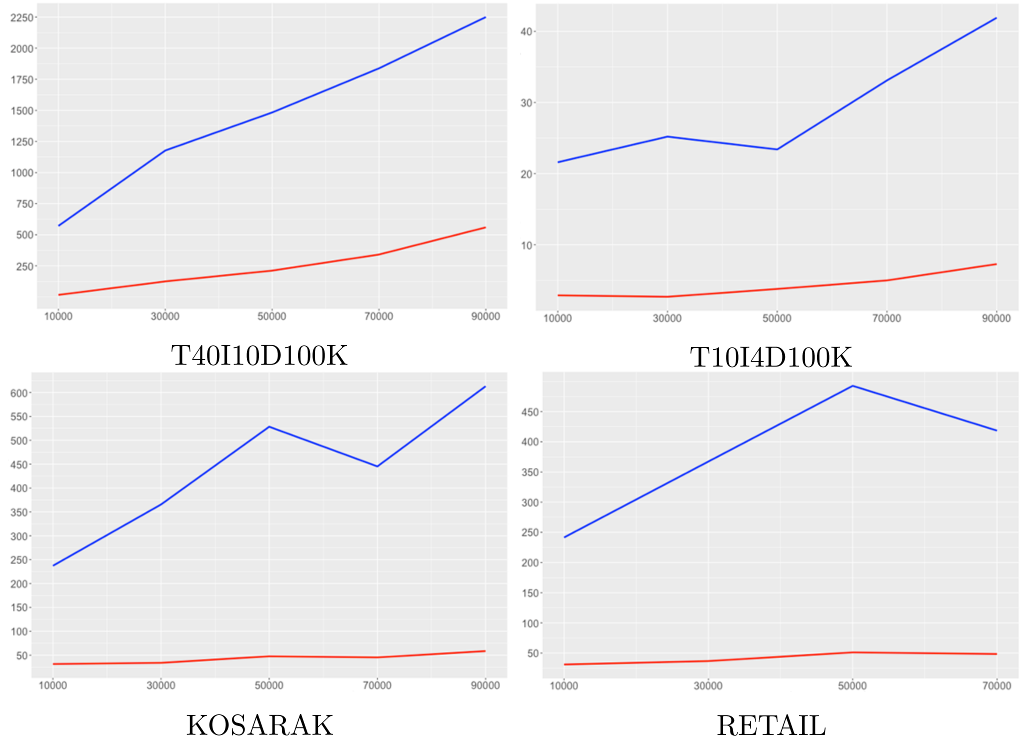
Among the chosen datasets we have selected those used in [LZZ$^+$14] in order to recreate the results of FIMoTS. These datasets present different features: wide/limited number of transactions and itemsets, low/high correlation of transactions with respect to the vocabulary (items), different length of transactions, synthetic/real data, etc. Datasets T10I4D100K and T40I10D100K have been artificially generated by the IBM tool, KOSARAK comprises data about the clicks made in a news portal from Hungry, and RETAIL contains information about the purchased products in a Belgian commerce.

Figure 25 shows the time spent by original FIMoTS and our distributive proposal for the different datasets using different window lengths (x-axis). It is worth to mention that for the KOSARAK and RETAIL datasets the process did not conclude, in the non-distributed case, for the larger windows (number of transactions) due to a memory overflow. This is mainly due to the large number of items in these datasets. It is important to note that in all cases the distributed version outperformed the original FIMoTS, as expected.

Regarding the memory consumption, Figure 26 shows the memory usage for the synthetic datasets (since for KOSARAK and RETAIL some experiments were not concluded). In the left graph, the behaviour of our approach is worst when increasing the window length. This is due to the replication phase where, the memory consumed by each node is higher than the quantity employed in the non-distributed FIMoTS. On the contrary, in the right graph we can observe that the memory consumed by the distributed version improves the memory consumption in all cases, although the improvement is lower and it decreases as the window size increases.

Concerning the performance of our proposal we have carried out several experiments for different configurations of the *minsupp* threshold and analysed the time taken and the memory consumed in the initialisation phase and for the updating of the results in average for all the windows. For being able to compare the results we have fixed the window length to 10000 transactions. In tables I.3 -I.4 we can observe the results obtained in each of the datasets. From these results we can see that for the initialisation tree process the spent time is higher than its updating when a new window is considered. This is because the algorithm must compute all the transforming bounds and supports of itemsets, while during the updating phase only part of the tree is recalculated. Moreover, the execution time spent during the updating process remains more or less constant in each incoming package even when the quantity of itemsets is very high (see the cases for the minimum support of 0.01). Finally, the memory employed in each case remains proportional to the quantity of itemsets processed.

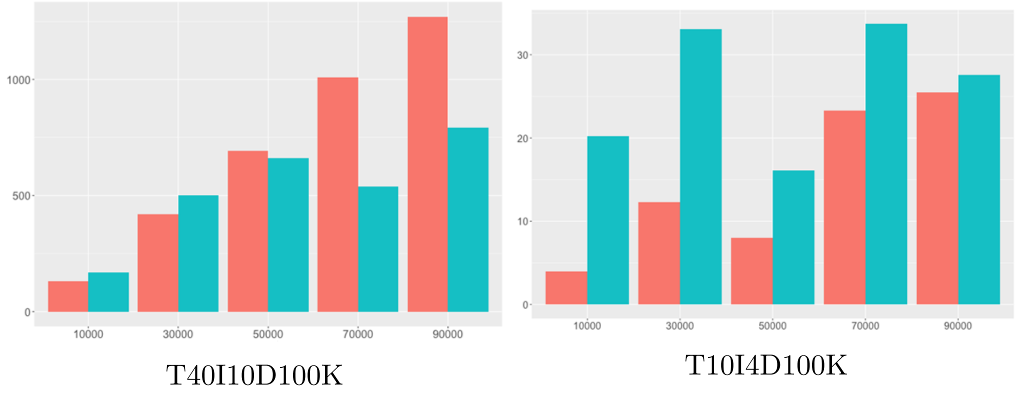
The developed proposal is intended to enable the analysis of tendencies in massive amounts of data coming from data streams. This is extremely useful in areas where the amount of generated



T40I10D100K T10I4D100K

KOSARAK RETAIL

Figure 25: Average time in seconds (y-axis) for different window lengths (x-axis). Red line represents our proposal and blue line the non-distributed FIMoTS



T40I10D100K T10I4D100K

Figure 26: Memory consumed in MB (y-axis) for different window lengths (x-axis). Red colour represents our proposal and blue colour the non-distributed FIMoTS

| *minsupp* | initialisation /updating packages | frequent/ infrequent itemsets | initialisation/ average updating time | initialisation/ average updating memory |
|---|---|---|---|---|
| 0.01 | 10 / 90 | 780 / 126098 | 931 / 853.66 s | 25.39 MB/ 2.53 GB |
| 0.05 | 10 / 90 | 20 / 938 | 22 / 6 s | 128.4 KB/12.84 MB |
| 0.1 | 10 / 90 | 1 / 866 | 12 / 2.7 s | 40.11 KB/ 4.01 MB |

Table I.3: Dataset T10I4D100K with 10000 transactions

| *minsupp* | initialisation /updating packages | frequent/ infrequent itemsets | initialisation/ average updating time | initialisation/ average updating memory |
|---|---|---|---|---|
| 0.01 | 10 / 79 | 443 / 18261 | 92 / 253 s | 8.96 MB/797.44 MB |
| 0.05 | 10 / 79 | 37 / 8609 | 30/ 33.3 s | 4.04 MB/359.56 MB |
| 0.1 | 10 / 79 | 21 / 8612 | 31/ 30.8 s | 3.97 MB/354.09 MB |

Table I.4: Dataset RETAIL with 10000 transactions

information exceeds the usual bounds of static databases and is in continuous movement. These are the cases of social media (twitter, linkedin, instagram, etc.) or economical/business analysis. In this work we focus in frequent itemset mining for finding the most relevant tendencies in data taking into account their appearance. We have revised the existent algorithms in frequent itemset mining taking into account both perspectives: massive data and data coming from streams. The best to our knowledge there does not exist an approach for frequent itemset mining taking into account both premises.

We therefore have developed an algorithm capable of extracting frequent itemsets in continuous flows of data by using the MapReduce framework and the Spark Streaming platform. The experiments carried out show that, as it was expected, our proposal outperforms the non-distributed version of the algorithm, and moreover, some experiments which could not be finished by the original FIMoTS can now be executed.

# 6 Association rules visualization

Data mining techniques are nowadays highly useful and widely used in industry, business and government. However, their broad adoption is sometimes limited because non-expert users are not able to rightly interpret and deal with the complex results obtained. One of the fundamental parts of the KDD process is that of decision making and knowledge exploration. Therefore the algorithms have to obtain interpretable results that are understandable to the end user.

For these reasons, in this section we present a methodology for visualizing association rules using an intermediate form to improve the interoperability. Moreover, with this methodology we also enable a faster processing of rules that better adapts to our necessities before choosing the visualization technique. The papers associated with this part can be found in Section 4.1 and 4.2 of Chapter II.

## 6.1 Background

In the literature, we can find several display applications [DC19, GBOZ18, KMCG17], in particularly for association rules [Li15, KT03]. However, all of them have a lack of interoperability due to a missing formal representation of association rules. For this reason, in this thesis we present a methodology for visualizing association rules using an intermediate form to improve the interoperability. Moreover, with this methodology we also enable a faster processing of rules that better adapts to our necessities before choosing the visualization technique.

This section is devoted to present basic concepts and terminology from visualization and data structure models, and review previous works and applications for visualizing association rules. We also explain the principal advantages and drawbacks of these tools and analyse their interoperability and interpretability. To this end, the different tools found in the literature have been classified in two different ways: 1) according to the type of visualization they enable to perform, and 2) depending on their functionalities (i.e. if the tool enables only visualization or if it also includes the frequent itemset or association rule discovery process).

### 6.1.1 Classification by type of display

Visualization of association rules can be used in different ways depending on the capabilities implemented by the tool, or by the type of visualization they implement. However, we must also take into account the structure of the results obtained and what the user wants to observe in the graphical representation in order to facilitate the results interpretation and to discover hidden information. These techniques can be classified in different subgroups according to the different ways and structures used to represent association rules. Consequently, we analyzed the tools for visualizing association rules attending to different categories, such as the data structure or the kind of visualization. We then classify the available visualization tools into several groups: tabular representation, parallel coordinates, visualization using matrices, trees, and graph-based methods (see Table I.5).

| **Name** | **Type visualization** | **Type by capacities** | **Focus** | **Advantages** | **Disadvantages** |
|---|---|---|---|---|---|
| Arules Table [Hah17] | Tabular 2D | Library<br>Methodology | Measures<br>Rule | -Intuitive<br>-Easy to use<br>-Show a wide variety of measures | -Problematic with many rules<br>-Limited capacity with rules with many items<br>-No overall view |
| CristalClear [OONL02] | Tabular 2D | Visualization | Measures<br>Rule | | |
| WiFIsViz [LIC08] | Tabular 2D | Visualization | Freq. itemsets<br>Rules | | |
| PEAR [JPA08] | Tabular 2D | Library<br>Methodology | Measures<br>Rule | | |
| Arules Scatter plot [Hah17] | Tabular 2D | Library<br>Methodology | Measures<br>set of rules | | |
| Mosaic plots [HSW00] | Tabular 2D | Visualization | Freq. itemsets<br>Rules | -Better compression of the relationship between items<br>-Possibility of using colors to visualize measurements | -Complexity in using rules with many items<br>-Only for reduced sets of rules<br>-No overall view |
| Arules Tow-key plot [Hah17] | Tabular 2D | Library<br>Methodology | Rule length | -Simplicity<br>-Ease to observe relationships between items<br>-Easy and intuitive understanding without knowledge | -Difficult to represent the measures of the rule<br>-Problems when increasing items or number of rules<br>-Better for rules with few related items |
| Parallel Coordinates [Yan05] | Parallel Coordinates | Visualization | Measures<br>Rule | | |
| Parallel Coordinates [BD08] | Parallel Coordinates | Visualization | Measures<br>Rule | | |
| Parallel Coordinates [HC11] | Parallel Coordinates | Library<br>Methodology | Measures<br>Rule | | |
| Arules Grouped Matrix [Hah17] | Grouped Matrix | Library<br>Methodology | Measures<br>set of rules | -Displaying Large Rule Sets<br>-Useful for rules with 1 consequent | -It is not possible to display rules with several items in the consequent<br>-Not useful for sets with very Freq. itemsets |
| Grouped Matrix [UHB01] | Grouped Matrix | Library<br>Methodology | Rules | | |
| CristalClear [OONL02] | -Graph<br>Hybrid:-Chord<br>-Tree | Visualization | Measures<br>Rule | -Observation as a whole and on concrete rules using various methods<br>-Interactive<br>-Selection and filter combining methods<br>-Diversification of measures to improve the exploration of the results | -For specific use cases<br>-More complex management tools with many filters and interactions between displays<br>-Some cases it is necessary to have certain data such as geolocation, temperature data because the displays are developed for those data. |
| Visual discovery of network patterns [SG08] | Hybrid:<br>-3D Matrix<br>-Bar diagram | Visualization | Freq. itemsets<br>Rules | | |
| Hierarchical visualization [CXSP17] | Hybrid:<br>-Chord<br>-Graph | Visualization | Freq. itemsets<br>Rules | | |
| Criminal visualization [XC05] | Hybrid:<br>-Chord<br>-Graph | Visualization | Freq. itemsets<br>Rules | | |

| **Name** | **Type visualization** | **Type by capacities** | **Focus** | **Advantages** | **Disadvantages** |
|---|---|---|---|---|---|
| Arules Matrix [Hah17] | Matrix (2D, 3D) | Library Methodology | Rules | -Facility to explore items by measures<br>-Possibility to visualize measurements by shapes, colors | -Difficult to scan itemsets in a rule<br>-Problems when visualizing sets of many items (insufficient axis length)<br>-Overview of the whole (interactive tools improve this point) |
| Matrix [BJA99] | Matrix (2D, 3D) | Library Methodology | Rules | | |
| Matrix for text mining [WWT99] | Matrix (2D, 3D) | Library Methodology | Rules | | |
| Arules interactive [SPH$^{+}$17] [Hah17] Matrix | Matrix (2D, 3D) | Library Methodology | Rules | | |
| ArulesViz [Hah17] | Graph | Library Methodology | Items | -Wider vision as a whole<br>-Possibility to apply different layouts<br>-Ability to view large rule sets<br>-Interactive<br>-Ability to view rules consistent with various items<br>-Use of color, shape, directionality on edges and nodes to express rule measurements | -When we work with large sets of rules are produced many nodes and edges so it is necessary to select a good algorithm of layouts for the visualization and interpretation are correct.<br>-Not prepared to handle different measurements at the same time<br>-It is more complex for the user because it is necessary to know how is the structure of a network and how are represented the rules in it.<br>-Some of them are difficult to interpret |
| VisAR [Yan03] | Graph | Visualization | Freq. itemsets Rules | | |
| Criminal Network [XC05] | Graph | Visualization | Freq. itemsets Rules | | |
| Product network analysis [KKC12] | Graph | Visualization | Freq. itemsets Rules | | |
| Projection Explorer [LPPM07] | Graph | Visualization | Freq. itemsets Rules | | |
| Spanning trees [VRM18] | Graph | Library Methodology | Freq. itemsets Rules | | |
| SYSNETS [SSPB16] | Graph | Visualization | Freq. itemsets Rules | | |
| RDViz [LC10] | Graph | Library Methodology | Rules | | |
| WiFIsViz [CXSP17] [LIC08] | Graph | Visualization | Freq. itemsets Rules | | |

Table I.5: Comparative table of tools for visualizing association rules.

### 6.1.2 Classification by type of capacities of the technology

In order to visualize association rules, different types of tools can be found, depending on whether these only enable the rules visualization or to carry out the extraction process of association rules. According to this, we can classify them into two groups. On one hand, there are tools that work as libraries that allow the extraction of frequent itemsets, association rules and the visualization of the results (this process flow can be observed in Figure 27). On the other hand, there are tools that allow loading the results obtained from other tools and visualize them (this process flow can be observed in Figure 28).

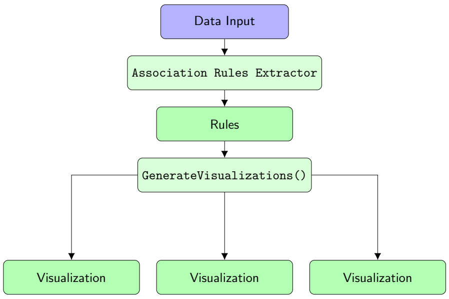


Figure 27: Pipeline with rule extraction included

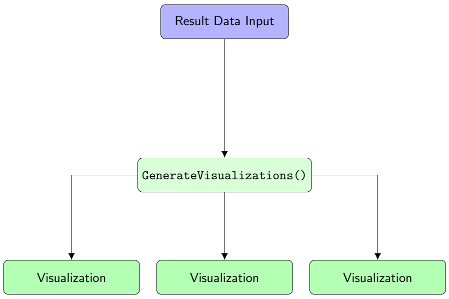


Figure 28: Pipeline using only visualization

Within the first group we find different methods that allow us to load the results obtained with association rules extraction tools. These must be loaded using a different standard structure depending on the tool for being able to visualize them. Among them we can highlight several tools [HSW00, BD08, Yan08, WWT99, SG08] that, from the stored results as association rules, allow to visualize the rules in different ways.

In the second group we find methods that enable complete analytical procedures from the same tool. For example in the Arules package [HC11] we can extract frequent itemsets and association rules to later visualize them making use of this library. PEAR works in a similar way, it follows the methodology developed in [JPA08] allowing to perform the whole process, from the loading of data to the visualization. Its main difference is the use of a set of rules, where through the use of operators, we can to unify these rules by grouping them into sets. This new set of rules is displayed with the available methods.

## 6.2 Development

All the tools reviewed in work of Section 4.1 Chapter II have different advantages and disadvantages. As we can see from them, there are many tools that allow to visualize association rules. However, all these techniques are independent of each other using different formats for the rules extraction and for the use of results to display them. In addition, the methods that enable the complete process from the rules extraction to the visualization have some problems because they do not allow to use other methods for visualization or extraction. That is why we propose a methodology using an intermediate form with the purpose of improving the interoperability, being able to use different visualization methods by transforming the rules into an intermediate form that can be used in a wider range of visualization tools.

On the other hand the intermediate form proposed gives us a format to store the results of the rules that allows to consult and efficiently filter by means of the elements and measures associated to the rules. For example, it is possible to store it natively in NoSQL databases, from which we can then display them by using visualization libraries such as *bokeh, arules* or *3D.js*.

Finally, this methodology is generic because it is applicable in tools such as those that use a workflow like the one shown in Figure 27, performing the extraction of the rules and then the visualization. This improves the use not only of the visualizations that a tool implements, but also being able to employ the intermediate form to use visualizations of other tools. On the other hand, in the case of the tools following the pipeline of Figure 28, this enables a generalization of a visualization technique for any association rules extraction algorithm

Our proposed methodology is based on a transformation of the rules into a graph, adding different capabilities to improve their display. The procedure we followed to make a visual representation of the association rule (depicted in Figure 29) can be divided into three phases. The first one is in charge of obtaining the rules by means of association rules mining algorithms. Next, we transform and process these results into a graph representation. The final step is the visualization of rules using different tools by means of this representation.

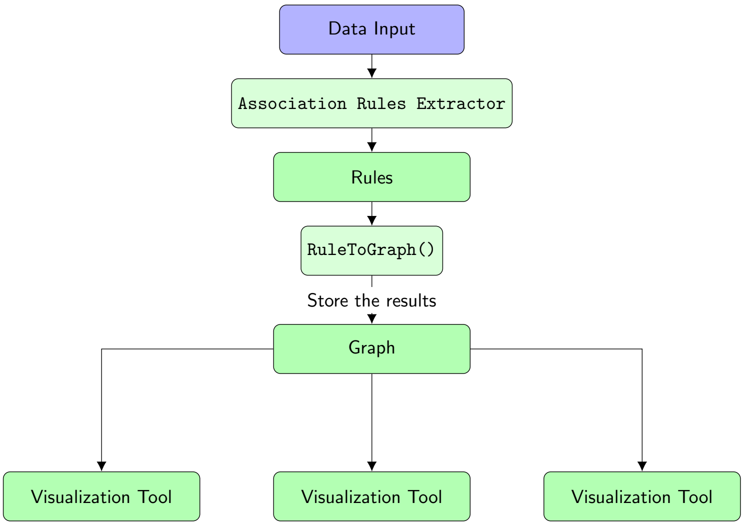


Figure 29: Pipeline proposed for association rules visualization

### 6.2.1 Association rules transformation into a graph

The first step before visualizing the rules is to transform them into a graph. There are different options to do so. In our case, we make a transformation to a graph comprised of two types of nodes: items and rules. The nodes will be connected according to the results that have been obtained. Figure 30 shows an example of rule transformed into a node which connects the item-type nodes.

The transformation is performed for each rule of the database, and carried out taking into account the corresponding associated characteristics according to the type of node: rule or item. For example, for a rule type, the node will contain information about the measures of interest (e.g. support, confidence, lift). Then, the edges of the graph are added, taking into account the type of link (whether it is the consequent or the antecedent). When this process ends, the rules are represented by a graph through which we can make different types of display.

Figure 31 exemplifies the structure followed for representing an item node. In this case, the name indicates the pair 'atribute_value' that represents the item; the `group` and `kind` represent the

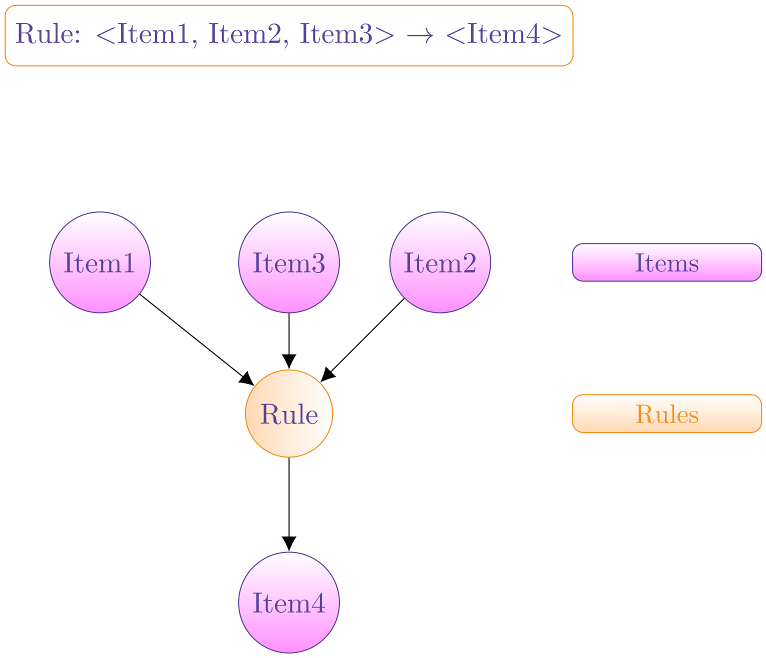


Figure 30: Example of transformation from rule to graph structure

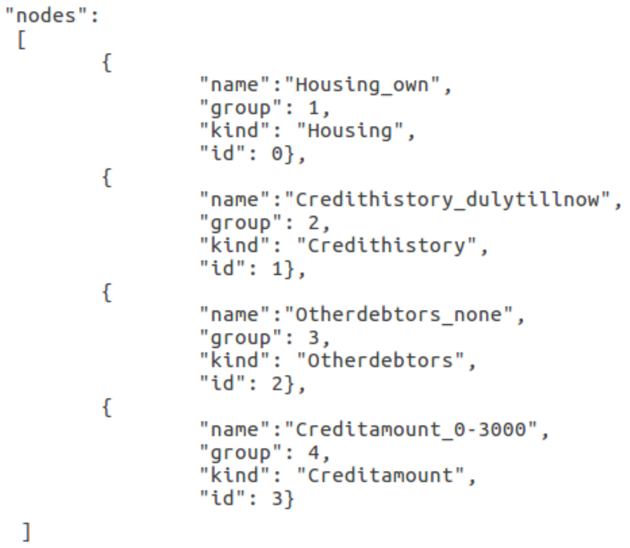
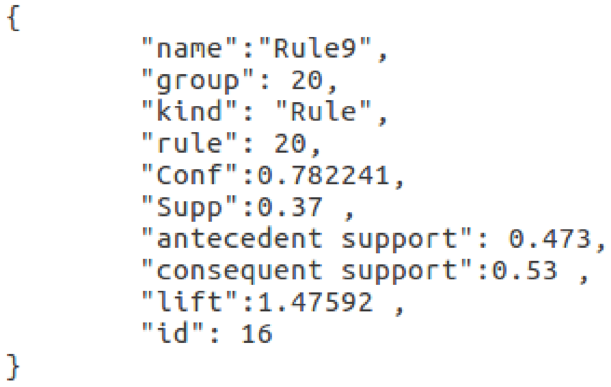

```
"nodes":
 [
        {
                "name":"Housing_own",
                "group": 1,
                "kind": "Housing",
                "id": 0},
        {
                "name":"Credithistory_dulytillnow",
                "group": 2,
                "kind": "Credithistory",
                "id": 1},
        {
                "name":"Otherdebtors_none",
                "group": 3,
                "kind": "Otherdebtors",
                "id": 2},
        {
                "name":"Creditamount_0-3000",
                "group": 4,
                "kind": "Creditamount",
                "id": 3}
 ]
```

Figure 31: Example of Json for item type nodes

attribute `name` of the item or whether it is a node of rule type (see Figure 31). The `group` number is used to improve the efficiency in some types of visualization such as graphs. Finally, `node id` is used for univocally identify the node.

On the other hand, in Figure 32 we can see the structure for nodes representing a rule in the graph. In this case, this type of element also stores the measures related to the rule such as the support, confidence, etc. This generated structure allows a later visualization of the obtained rules in a way that improves the interpretability and comprehension of the results.



```
{
        "name":"Rule9",
        "group": 20,
        "kind": "Rule",
        "rule": 20,
        "Conf":0.782241,
        "Supp":0.37 ,
        "antecedent support": 0.473,
        "consequent support":0.53 ,
        "lift":1.47592 ,
        "id": 16
}
```

Figure 32: Example of Json for rule type nodes

### 6.2.2 Visualization tools

Having transformed the rules to a graph structure following the methodology described in Figure 29, we can make use of the intermediate form to employ different visualization tools. Taking into account the above mentioned structure, we can use different graph visualization libraries such as Gephi [BHJ09], D3JS [BOH11], Bokeh [Bok14], ggnet2 [TBH17], NetworkX [HSSC08], NOESIS [MBGCT16], etc. These tools return a graphical display of the transformed rules. Thanks to the use of our methodology we can visualize rules with more than one consequent (for instance, the rule extraction algorithm of the Arules package only allows rules of only one consequent) using an external algorithm [FBRMB16] and using the intermediate form previously explained.

Moreover, we can use libraries and visualization tools to customize exiting visualizations tools to adjust according to user preferences. A powerful visualization package enabling this kind of modifications is 3DJS [BOH11]. For example, in Figure 33 we have adapted a graph-based display called Force directed graph, where a different color is used for each rule, and every node of type 'Rule' shows the associated support and confidence of the rule. Additionally, the directionality of the arrows to the rule node indicates whether an item node is part of the antecedent (arrows pointing to the rule node) or part of the consequent (arrows pointing to the item node). A variation of the previous visualization is also made in Figure 34 where we have customized some properties of the graph, like for instance to indicate the item support in the edge. Another example is the chord diagram shown in Figure 35, also customized from making use of the 3DJS library. This visualization is interactive: when you are positioned in the items the edges are emphasized and colored according to the relationship they have with other items (antecedent or consequent). With respect to the efficiency and scalability of the visualizations on graphs, the developed tool is able to generate the graphs in SVG format (Scalable Vector Graphics) being able to be distributed by means of the `Tuoris` library [MFMSG20].

## 6.3 Results and discussion

In this research we have reviewed the most common and used methods to visualize association rules. From the different tools examined we have seen that each one comes with strengths and weaknesses. As part of this analysis, we have been able to see how there is no one display tool that works well for all cases. This is because based on the problem, the data set and the objective, we need to

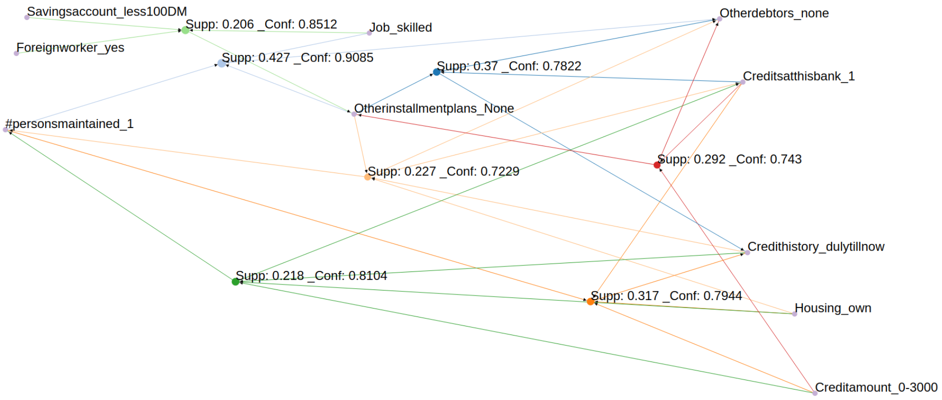


Figure 33: Example developed using D3JS through the intermediate form

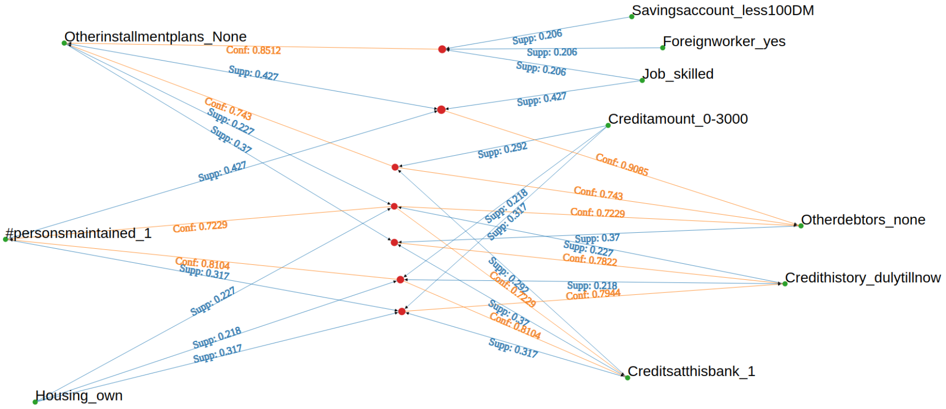


Figure 34: Example developed using D3JS through the intermediate form

choose the better visualization solution according to our requirements. In addition, some of the tools allow to export data from our own algorithms and others do not, therefore it is necessary to have a standard format with which to exchange our results among the different tools in order to exploit the strengths of each tool, adding interoperability to the process. Thus, in this way we can have more resources and not depend on a single tool to get a better visualization and understanding by the end user. Therefore our proposal allows us to extract the association rules from any algorithm or from a library and then export it to be displayed employing the different available libraries: 3D.js, Arules, Bokeh... Furthermore, this standard and widely known format can be stored natively in databases that allow to efficiently filter, search and store large sets of results.

In summary, we have carried out an extensive analysis of the different visualization tools available in the literature, as well as classifying them according to the type of visualization and the capabilities they offer. After this analysis we have seen that most of the tools only allow to visualize rules from a predefined structure, or the library containing the visualization only allows the generation and display of rules through their own algorithms. Therefore, in these cases, the user cannot combine their own or other available association rules mining algorithms with available techniques to present association rules in a graphical way.

Because of it, in this research we put forward an intermediate form which allows to use a great variety of the tools for association rules, and in addition, it allows to combine them with the different available visualization techniques.

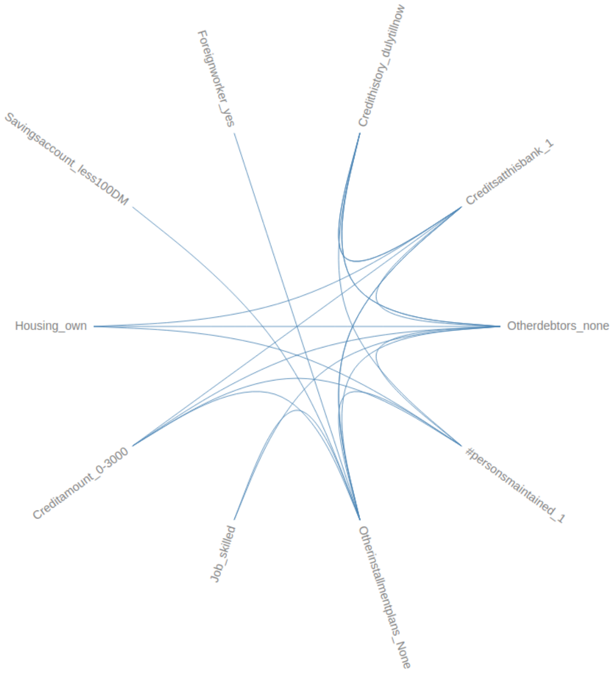


Figure 35: Example of visualization by means of a chord diagram

# 7 Applications

In the previous sections we have studied and developed algorithms for knowledge extraction using different techniques such as fuzzification, fuzzy association rules, frequent itemsets and visualization in Big Data environments. In order to validate and demonstrate these techniques in real environments, two applications have been carried out in two different sceanrios. On the one hand the application of a method to improve energy efficiency by means of a novel control algorithm that uses a simulation model to optimize the operation of heating, ventilation and air conditioning in non-residential buildings. On the other hand, a complete application has been developed for the extraction of fuzzy association rules from sensor data, meteorology and occupation of non-residential buildings. This section describes both applications.

The papers associated with this part can be found in Section 5 of Chapter II.

## 7.1 Background

Companies in the energy sector are increasingly aware of the great opportunity that the analysis and exploitation of these data can bring (see for instance [KÖGP12, NÖW19]). To this end, there is a tendency to store and process this type of data in order to obtain a meaningful insight into consumption patterns and the operation of equipment.

However, the enormous amount of data generated by sensors and other data sources, such as meteorological and occupancy data, require new infrastructures and algorithms capable of storing, processing and analysing them. Big Data presents great opportunities for implementing new solutions to manage these massive data sets. Moreover, the nature of these data can be diverse and can be described in numerical, categorical or imprecise forms. To improve the interpretability of the data we can modify knowledge extraction algorithms through the use of fuzzy logic to create, for example, linguistic labels for sensors with numerical values that also provide meaningful semantics for the user.

The main challenge when processing large energy data is therefore to provide adequate methods and techniques capable of improving the quality of the data generated by the sensor metering of buildings. In this regard, different methods can be applied during the pre-processing phase in order to detect outliers [YAW16] or by applying other cleaning data procedures. In general, these data are massive because they are generated with low frequencies by a large number of sensors. This massive quantity gives grounds to use Big Data techniques to process the data in a distributed way, thus improving the efficiency of the processes.

Sensor metering data are very often collected by means of numerical measurements taking values within a continuous range. This increases the difficulty of analysing them on a larger scale due to their fine granularity. A primary approach is to divide the range of possible values into intervals in order to help the algorithms to process the data. But this division suffers from some drawbacks: firstly, the results can vary a lot depending on the applied division, and secondly, this division may not be very intuitive for its later analysis of results. Fuzzy sets have been proven to adequately represent data with soft borders, increasing the interpretability of results by associating meaningful linguistic labels to the generated fuzzy sets. Other approaches use interval programming methods to tackle the uncertainty that the data may contain [MR17].

On the other hand, besides using the information about of the behavior of the buildings, another possibility is to assist in the decision making process employing the extracted knowledge joint with expert knowledge for the saving of energy.

In the last decade, several proposals for automating the generation of operational plans based on

Model Predictive Control (MPC) have been presented [AJS14, MD16]. MPC uses a simulation model of the building to capture its dynamic characteristics and predict its response to alternative control scenarios. It pursues a (conflicting) dual target: reducing energy consumption thanks to pre-emptive control and anticipation of the building state while keeping users' comfort. By establishing a complete sequence of instructions for the building equipment i.e. the (daily) operational plan–, it overcomes the limitations of homeostatic controllers, which cannot guarantee long-term optimal operation: the ahead time and the timespan of the control instructions can expand to several hours, leading to plans entailing more uncertainty because of the use of forecasted building conditions (e.g. weather, occupancy) and more complexity because of the exponential increase of possible plans, but also more efficient because of the exploitation of the inertial effects of HVAC equipment.

In this thesis we propose two applications, on the one hand a fuzzification algorithm to adequately pre-process the data in order to apply, in a later step, fuzzy data mining techniques to discover potentially useful information that may be hidden in the data. And on the other hand a control system capable of parameterizing a building management system to improve its energy efficiency.

## 7.2 Development

### 7.2.1 Probabilistic Algorithm for Predictive Control

The core of the system is the intelligent operational plan generator (OPG) module, an MPC-like control scheduler supported by a cloud-based extension of the IESVE2 [IES] simulation software. The OPG algorithm calculates an operational plan (OP) for a future period (typically the next day) after simulating hundreds of candidate plans under the forecasted state of the building (i.e. considering weather and occupancy estimations) in order to minimize energy consumption while guaranteeing occupants' comfort. Eventually, the OP setpoints are automatically applied to the equipment without direct involvement of the operator. To the best of our knowledge, this is the first proposal using an off-the-shell full-complexity model for predictive control. In particular, we have applied association-rule discovery, an unsupervised technique able to find existing relationships between variables and their values. In addition, these results allow the operators to carry out procedures and the use of the equipment in the building with a better performance, thus improving its energy efficiency. On the other hand, the use of weather and occupation patterns for the energy use of the building would allow the optimization of the needs of the building in specific situations. The Operational Plan Generation module is the core of the control system in the Energy IN TIME project. It encompasses three main stages (see Figure 36):

1. Collection of forecasted building data: The OPG retrieves the weather forecast and the occupancy prediction for the operation period, usually the next day. Within our project, the weather forecast was obtained from the Weather Analytics, and occupancy predictions were obtained by using an agenda, which identified working days and average room occupancy per hour.

2. Generation, simulation and evaluation of candidate plans: The OPG runs several simulations to reproduce the expected building behavior, in terms of energy consumption and comfort, under different operation plans and according to the forecasted conditions retrieved in the previous stage. The best plan in terms of energy efficiency and comfort satisfaction is selected.

3. Storage and execution of best plan: The best plan is stored in a database and made it available to the setpoint writing component, which eventually sends the OP control instructions to the

BEMS. This database also stores the context associated to each selected OP, i.e. the forecasted building data used by the OPG algorithm and the simulation results. This information is useful to explain the rationale of an OP to the building managers, who can revise and modify the setpoint values in real time as well.

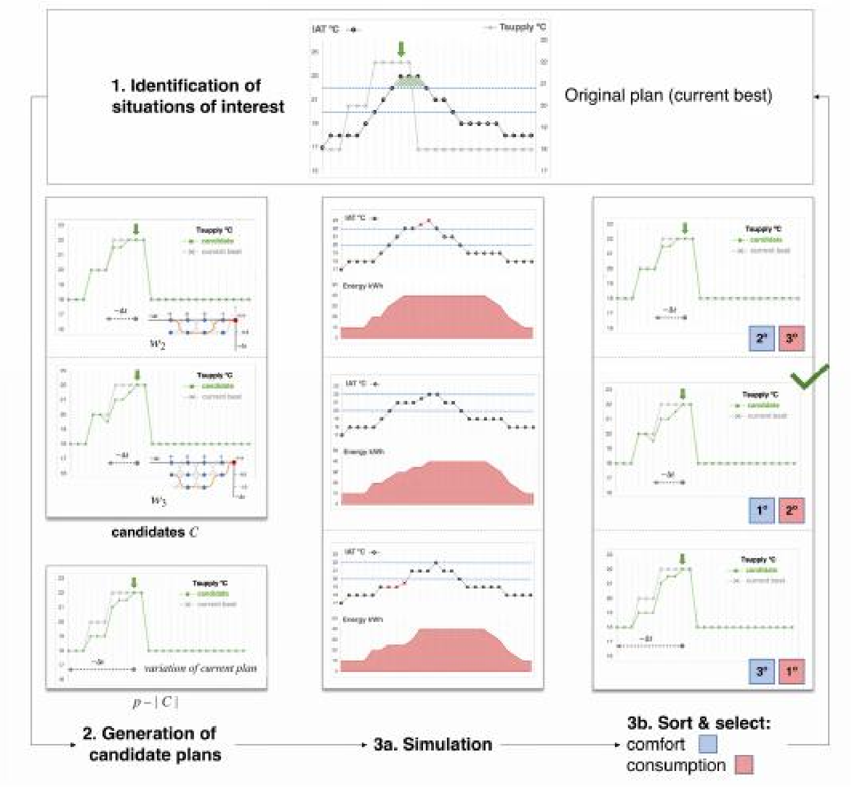


Figure 36: Overall functioning of the OPG algorithm, including main stages

### 7.2.2 A fuzzy mining approach for energy efficiency

The discovery and exploitation of information collected from buildings has attracted attention in the last decade due to its economic and environmental impact. Big Data offers a suitable framework for the efficient implementation of analysis techniques capable of handling large amounts of data, especially those produced in building management systems. In addition, the use of fuzzy logic can improve the interpretability of collected sensor data, offering improved results and interpretation to end users.

In this study, a data mining methodology has been implemented using the Big Data framework and applied to different data sets collected from an office building in Romania. In particular, we have applied a fuzzification algorithm to improve the application of data mining techniques such as association rules. The whole system has been deployed using the Spark platform to enable the analysis of such an amount of data generated by the sensors in the building. This technique has allowed the exploitation of different kinds of data collected in the diverse pilot areas of the building. The proposed solution has been applied to the static data collected from the building, obtaining different relationships that show the energy behaviour of the building and make evident some patterns that can be used to improve the energy efficiency of the building.

Figure 37 depicts the complete system of our proposal. It can be divided into three big blocks. In the first step, the data collection is carried out. Depending on the building, different types of data are generated with more or less reliability. For example, in the case of non-residential buildings (the example of our case study) the agenda containing the occupation of the building will be more reliable than that of a hotel, where the guests do not have fixed schedules.
The second step comprises the core of the computation with different phases: pre-processing, fuzzification and application of data mining techniques. In our approach, distributed processing tools are used to improve efficiency and computational capacity.
Once we have obtained the results of our analytical techniques, the last step is the interpretation of results by experts or operators of the system in order to obtain knowledge that may be useful for improving the maintenance processes and the energy efficiency of the building.
This whole process has been applied in the field of the Energy IN TIME project to non-residential buildings: a hotel, an airport and two office buildings, although we just present here the results obtained from one of the office buildings. Nevertheless, the proposed system is general and can be applied to other types of buildings.
To this end, we have deployed the whole system following the Map-Reduce paradigm which allows the distributed computation of large volumes of data. Specifically, we used the Spark platform [KKWZ15] together with an unstructured storage following NoSQL specifications, which enables an efficient storage of sensor data collected in buildings.

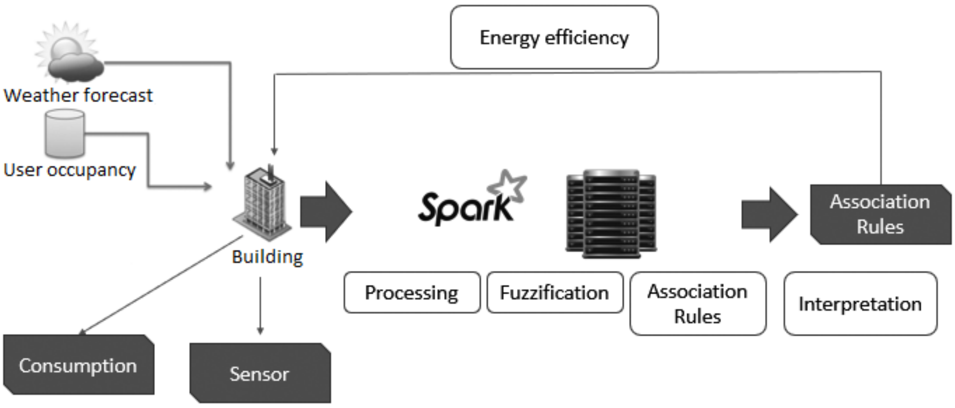


Figure 37: General process of our proposal

In our proposal, before applying fuzzy association-rule mining, we have to pre-process the data collected from the building. We can observe the workflow of the proposal in Figure 38, where we have distinguished several phases.

The first phase comprises the collection and storage of building data which are data from sensors and building equipment, weather and occupation.
Subsequently, the pre-processing phase of the data is carried out. The collected data from sensors may contain lost values, outliers and so on and therefore some transformation before to be used by the algorithms is needed.
After this, the continuous numerical values are transformed following a fuzzification method. By means of this process the range of values are divided into meaningful fuzzy sets where some linguistic labels can be associated to each fuzzy set. This step improves the interpretability of the data and the obtained results will adjust better to the nature of the variables. Depending on the type of variable, the fuzzification procedure can either be carried out automatically or by the help of expert knowledge.

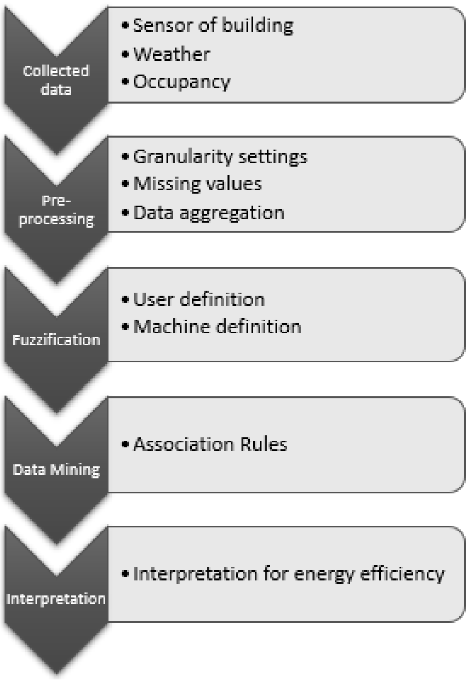


Figure 38: General workflow of the proposed data mining framework

Finally, when we have pre-processed and transformed the data, we proceed to apply data mining techniques. In this case study, extraction of fuzzy association rules has been applied in order to extract hidden relationships from the sensors, occupation and the environment of the building. Once the results have been obtained, the discovered patterns are interpreted with the help of end users to improve the energy efficiency of the building.

## 7.3 Results and discussion

### 7.3.1 Probabilistic Algorithm for Predictive Control

This research has presented the design and the implementation of an MPC-based control system aimed at reducing energy consumption in non-residential buildings while guaranteeing occupants' comfort. The main difference of our proposal with respect to other approaches is that we use a full-complexity simulation model, which runs in parallel in the cloud. This allows using more accurate models and facilitates communication between computer scientists, building operators and simulation developers, exploiting synergies of their joint work. Comprehensive quantitative and qualitative comparison with MPC approaches using reduced-complexity simulation models would be useful to support decision-making between different alternative approaches. Figure 39 and 40 show the comparison of the values of daily energy consumption in the pilot area obtained from the BEMS with the values estimated by the prediction models for the test period in the real building. Figure 41 shows the energy savings achieved in % of the (estimated) consumption before optimization.

As summarized in Table I.6, the average savings per day are, respectively, around 40% for the thermal subsystem and around 20% for the electrical subsystem. Weekends and holidays offer opportunities for higher energy savings, since the OPG adjust the operation to the building occupancy better than the manual operation. Savings have been achieved without compromising users' comfort.

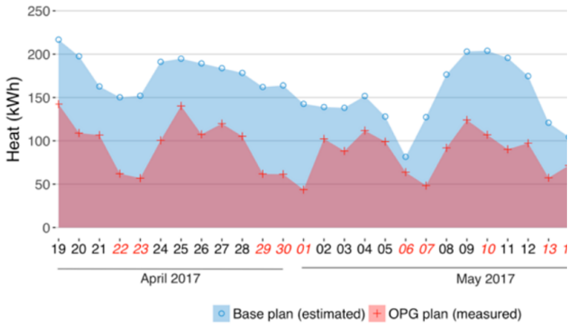


Figure 39: heating

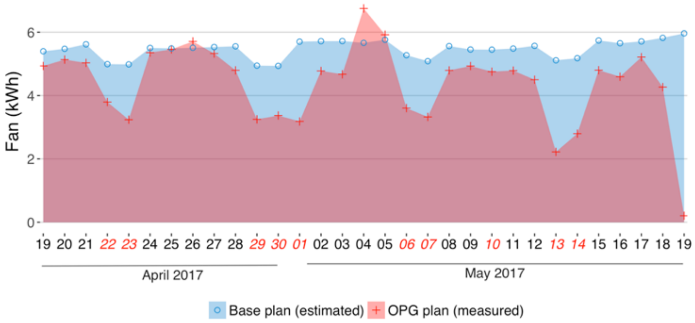


Figure 40: VAV fans

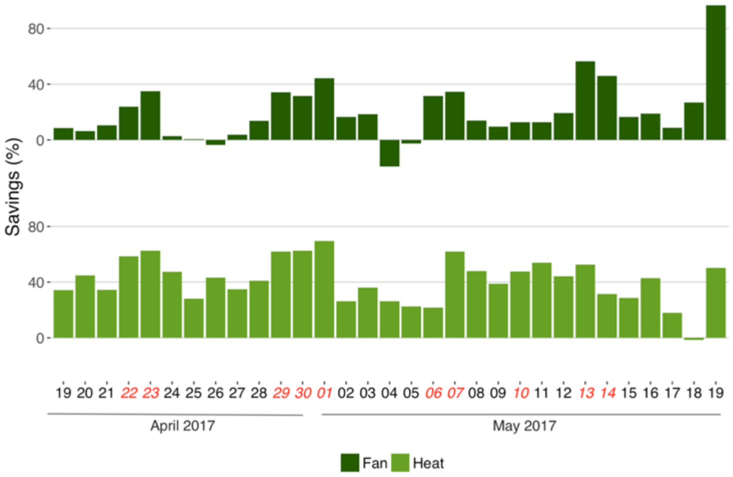


Figure 41: Savings

Comparison of daily energy consumption (kWh) during the test period vs estimated by the baseline models (Days in red italic font are weekend or holiday days)

Our system reduced the temperature setpoints given by the normal operation of the building between 0.5 and 2 °C. During the on-site test, this meant savings in heating above 40% (Table I.6) while keeping comfort. The algorithm adapted well to workdays and weekends, showing slightly better results in the former ones Figure 41. A possible explanation for this is that operators have lower availability in weekends and holiday days, and therefore it is not possible for them to create customized plans. The airflow consumption was also reduced in a 20% (Table I.6) without compromising the CO2 concentration comfort.

Experimentation in the Sanomatalo building, located in Helsinki, both in the simulation environment and in the real building, has shown that important energy savings (up to 40% at the end of the winter season) can be achieved, particularly by optimizing the control of the heating equipment.

### 7.3.2 A fuzzy mining approach for energy efficiency

Figure 42 shows some of the association rules found in an office building in Bucarest, taking into account their support and confidence. Figure 43 shows a sub-set of the discovered rules in the form of matrix, with the consequent (LHS) and the antecedent (RHS) of the rules.

| | **Overall** | | **Workdays** | | **Weekends & Holidays** | |
|---|---|---|---|---|---|---|
| | Heat | Fan | Heat | Fan | Heat | Fan |
| Estimated<br>(daily avg) | 156.47 | 5.47 | 164.70 | 5.62 | 132.79 | 5.06 |
| OPG<br>(daily avg) | 91.12 | 4.37 | 101.77 | 4.78 | 60.50 | 3.20 |
| Savings (daily avg %) | **41.76** | **20.12** | **38.21** | **14.91** | **54.44** | **36.73** |

Table I.6: Energy savings (kWh) achieved in the on-site test with the OPG control vs estimated by the baseline models

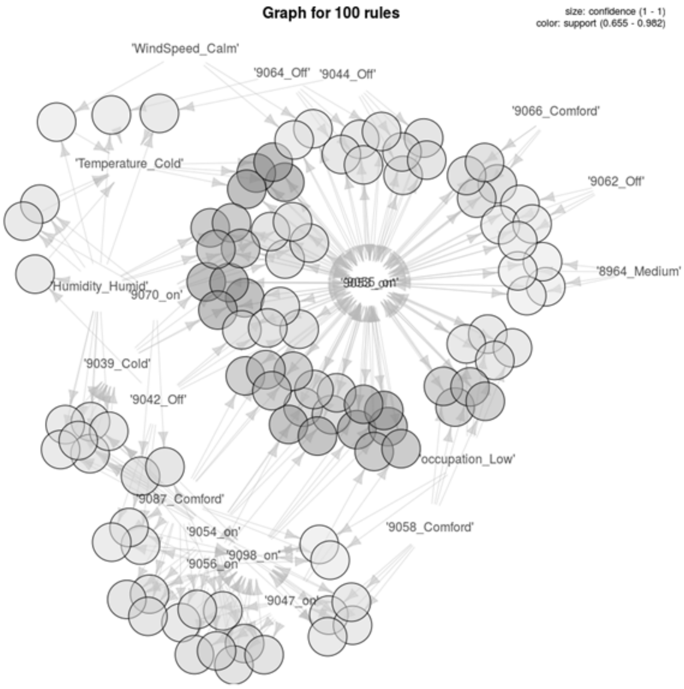


Figure 42: Graph visualization of some association rules discovered for the office building in Bucharest

The obtained set of rules has allowed us to discover hidden patterns in the operation of the building, which experts can then use to improve its efficiency and maintenance.
Having a look at the discovered patterns we can highlight different rules. For example, in Figure 43 the rule at the top left of the graph (position column 3, row 1) is:

$$\{9098 = on, 9039 = cold\} \rightarrow \{9061 = comfort, 9096 = cold\}$$

which changing the identifiers of the sensors to a more descriptive name results in:

$$\{Setup\,PAN = on, Output\,temperature = cold\} \rightarrow$$
$$\{PAN\,temperature = comfort, PAS\,temperature = cold\}$$

In this rule we can observe how the general operation of the building is described, i.e. outside the temperature is cold, the heating setup is ON and for that section of the building (PAN represents north area) the comfort temperature is achieved. In addition we can see that other sections of the building such as PAS (representing the south area) is cold at the same time, so we could determine that there are two rooms or sections that are not usually occupied at the same time.

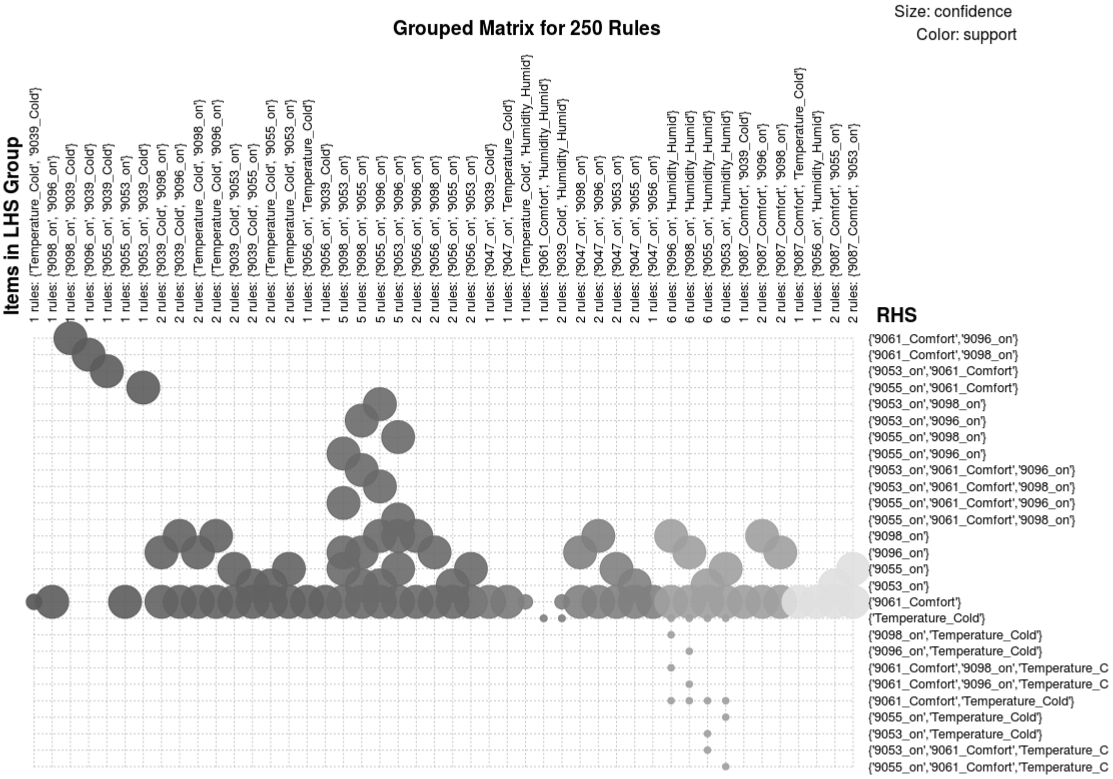


Figure 43: Some association rules discovered for the office building in Bucharest. LHS stands for Left Hand Side of the rule or Antecedent and RHS for Right Hand Side or consequent.

# 8 Concluding remarks

In this thesis, we have addressed several problems, focusing on a common goal: the analysis, design, development and evaluation of scalable fuzzy association rule extraction algorithms and the visualization of these results. In addition, these algorithms have been applied to real-world problems such as energy efficiency and streaming data.

to achieve the first objective of our research line, different innovative methods of association rules extraction in Big Data problems have been proposed. The experimental results have demonstrated the exceptional stability and efficiency, with respect to other classical methods.

The second objective proposed in this thesis was achieved by proposing several innovative methods for extracting fuzzy association rules from Big Data problems. The experimental results have demonstrated the exceptional stability and efficiency, with respect to other sequential methods.

In the third objective we focused on the extraction of frequent itemsets in data flow environments. The problem with streaming is the need to use very efficient algorithms for being able to analyse the data in real time analysing their trend over time. For this purpose, a frequent itemset mining algorithm has been developed which, through the use of sliding windows, is capable of updating frequent itemsets in a distributed way in data flows.

In the fourth objective we focus on the last phase of the KDD process, to improve the visualization of the results obtained by the association rules techniques. For this objective we have carried out a revision of the visualization techniques in the literature. In addition, we have proposed a visualization through an intermediate form that allows us to visualize the results by employing a variety of visualization techniques and tools.

In the fifth objective we wanted to address on one hand the pre-processing of data. For that a fuzzification method was proposed to improve data interpretability for end users, and we applied these techniques in a real use case with data from sensors of a non-residential building. On the other hand, the development of an application in a non-residential building to reduce consumption. A real case of use of this control method in a building in Finland has been presented with a result of great energy savings during its application.

## Conclusiones

En esta tesis, hemos abordado varios problemas, centrándonos en un objetivo común: el análisis, diseño, desarrollo y evaluación de algoritmos de extracción de reglas de asociación escalables y la visualización de estos resultados. Además, estos algoritmos se han aplicado a problemas del mundo real como la eficiencia energética y el análisis datos en streaming.

En el primer objetivo de nuestra línea de investigación, se han propuesto diferentes métodos innovadores de extracción de reglas de asociación en entornos de Big Data. Los resultados experimentales han demostrado la excepcional estabilidad y eficiencia, con respecto a otros métodos clásicos.

El segundo objetivo propuesto en esta tesis se logró proponiendo varios métodos innovadores para extraer reglas de asociación difusas en entornos de Big Data. Los resultados experimentales han demostrado la excepcional estabilidad y eficiencia, con respecto a otros métodos secuenciales.

El tercer objetivo se centra en la extracción de itemsets frecuentes en entornos de flujos de datos. El problema del streaming reside en la necesidad de usar algoritmos muy eficientes para poder analizar los datos en tiempo real y así poder analizar su tendencia en el tiempo. Para ello se ha desarrollado un algoritmo de extración de itemsets frecuentes que mediante el uso de ventanas es capaz de actualizar los itemsets frecuentes de forma distribuida en flujos continuos de datos.

El cuarto objetivo se centra en la última fase del proceso KDD, para mejorar la visualización de los resultados obtenidos por las técnicas de reglas de asociación. Para este objetivo se ha llevado a cabo una revisión de las técnicas de visualización en la literatura. Además se ha propuesto una visualización mediante una forma intermedia que nos permite representar las reglas de asociación para poder utilizar una gran variedad de técnicas de visualización.

En el quinto objetivos se ha querido abordar por un lado el preprocesameinto de datos para fuzzificarlos y mejorar su interpretabilidad para los usuarios finales y la aplicación de estas técnicas en un caso de uso real. Se ha presentado una aplicación de estas técnicas de fuzzificacion y extracción de reglas de asociación en Big Data con un conjunto de datos reales, en particular el de un edificio no residencial sensorizado. Por otra parte, el desarrollo de una aplicación en un edificio no residencial para reducir el consumo. Se ha presentado un caso real de utilización de este método de control en un edificio de Finlandia con el resultado de un gran ahorro energético durante su aplicación.

# 9 Future work

Based on the conclusions drawn from this thesis, new and promising lines of research can be proposed. They aim to improve existing models, and address new problems that are emerging from the Big Data evolutionary scenario.

- **Association rules mining in Big Data.** Our intention is to implement more efficient association rules mining algorithms by conveniently changing the PFP to extract all frequent itemsets. Additionally, we plan to apply the developed algorithms to extract patterns in sensored buildings to improve their efficiency behaviour.
- **Fuzzy Association rules mining in Big Data.** We also intend to generalise these Big Data procedures to other data mining techniques that use fuzzy association rules such as gradual dependencies [H02̈], [BCSV07], exception and anomalous rules [DRS11], etc.
- **Big Data Frequent itemsets mining in streaming.** In future works, we plan to apply the developed algorithm for social media analysis in real time and extend it to consider association rules mining in order to study the co-occurrences of frequent items in data streams.
- **Visualization of Association Rules.** Regarding future research, we aim to implement a complete open-source library for the extraction of association rules and their transformation to the intermediate form. In addition, our purpose is to implement it in a modular way to have APIs between modules and to enable the use of different association rules extraction algorithms with the intermediate form, facilitating therefore the connection with available visualization libraries.
- **Application of fuzzy association rules mining.** In [MRS14], decision support systems are used to improve the operation of building elements or equipment maintenance. Furthermore, some of them use rules obtained from the behaviour of the users [JGW10] to improve the building functioning. Therefore, using the results provided by our proposal a decision system could be implemented incorporating the real functioning of the building. The barrier that arises to automatically analyse continuously generated data needs to be dealt with. This leads us to propose a future improvement of the proposed system for handling such a continuous flow and to process it in real time conveniently. To do so, there are recently developed utilities within the Spark framework that enables the processing of stream data called Spark Streaming [Pra18, FBFAMBR19]. This extension will enable live data streams to be processed by dividing them into batches which can be then processed by the Spark mining algorithms.
- **Probabilistic method for improving energy consumption.** The OPG algorithm opens several opportunities for further research. The current design relies on a variant of heuristic search, which can be hard to scale up if several variables are to be optimized at the same time. In this regard, other search and optimization techniques could be applied. Specifically, genetic algorithms allow balancing diversification and intensification of solution search by adjusting their parameters. Moreover, self-configuration could be supported by machine learning techniques able to identify successful operation patterns from historical data, and to apply reinforcement learning to reward and reuse particularly efficient OPG plans.

## Chapter II

# Publications

# 1 Association rules mining in Big Data

## 1.1 Extraction of association rules with Big Data (Big Data congress)

- Carlos J. Fernandez Basso, M. Dolores Ruiz, Maria J. Martin-Bautista. International Journal of Design & Nature and Ecodynamics
  - Status: **Published**.
  - (SCIMago 2016): Q4 (373/544) .

# EXTRACTION OF ASSOCIATION RULES USING BIG DATA TECHNOLOGIES

CARLOS FERNANDEZ-BASSO, M. DOLORES RUIZ & MARIA J. MARTIN-BAUTISTA
Universidad de Granada, CITIC-UGR

## ABSTRACT

The large amount of information stored by companies and the rise of social networks and the Internet of Things are producing exponential growth in the amount of data being produced. Data analysis techniques must therefore be improved to enable all this information to be processed. One of the most commonly used techniques for extracting information in the data mining field is that of association rules, which accurately represent the frequent co-occurrence of items in a dataset. Although several methods have been proposed for mining association rules, these methods do not perform well in very large databases due to high computational costs and lack of memory problems.

In this article, we address these problems by studying the current technologies for processing Big Data to propose a parallelization of the association rule mining process using Big Data technologies which implements an efficient algorithm that can handle massive amounts of data. This new algorithm is then compared with traditional association rule mining algorithms.

*Keywords: Apriori, association rules, big data algorithms, data mining.*

## 1 INTRODUCTION

The vast amounts of data generated, stored and analyzed by organizations and companies, and by extension by private users, has given rise to a new phenomenon known as Big Data. Imagine any particular day and think about the millions of tweets that are published on Twitter, the countless messages sent via Whatsapp and the multitudes of users who visit Facebook and interact by uploading photographs. If these huge volumes of information were not enough, in the world today there are also vast numbers of sensors everywhere that collect information in real-time, such as for example GPS signals and the information generated by smartphones.

Big Data is a phenomenon in a constant state of growth. It is of great interest to companies and to users to be able to analyze this information and extract useful conclusions that will be beneficial in economic terms or to society as a whole.

In this paper, we will be looking at how best to perform this analysis. There are various methods for mining data which enable us to analyze and extract interesting information from datasets. These methods run into problems however when they are used to analyze vast amounts of data, becoming less efficient at processing and analysis.

To this end, we will be studying the association rules technique as a method for data mining [1]. We will be implementing these methods using specific Big Data techniques, that is, Hadoop [2] and Spark [3]. The results of our experiments will show that this method improves the efficiency of the algorithm in terms of time and memory when the number of transactions increases. However, when the number of items increases, the algorithm does not offer significant efficiency improvements in terms of execution time compared to traditional methods. Nonetheless, thanks to the fact that the Big

WIT CONFERENCES
This paper is part of the Proceedings of the International Conference on Big Data (Big Data 2016)
www.witconferences.com

ISSN: 1755-7437 (paper format), ISSN: 1755-7445 (online), http://www.witpress.com/journals
DOI: 10.2495/DNE-V11-N3-178-185

Data techniques offer greater memory capacity, substantial improvements can be achieved when it comes to managing the memory problems that arise when generating the item combinations to be analyzed in massive datasets.

## 2 PREVIOUS RESEARCH AND RELATED WORK

The aim of the Knowledge Discovery in Databases (KDD) process is automated extraction of non-trivial, implicit, previously unknown and potentially useful knowledge from large volumes of data [4]. This process is made up of a series of stages namely selection, preprocessing, transformation, data mining and interpretation.

Data mining is the most characteristic phase of KDD, which is why this term is often used as a name for the whole process. The objective of this stage is to produce new knowledge that is useful for the user, by constructing a model that is based on the data collected and describes the patterns and relations between the data. With this new knowledge, users can make forecasts, understand data better and explain past situations.

Data mining covers a wide range of techniques which can be classified into two main types: supervised techniques such as the randomForest [5] or boosting [6] classification methods and non-supervised techniques such as clustering.

In the literature, there are already various examples of the implementation of these methods with Big Data techniques. These algorithms are palpable examples of big datasets distributed in large groups of servers using Big Data frameworks such as Hadoop or Spark. Some of the methods implemented include RandomForest and K-means (clustering) within the official Spark MLlib library [3]. These methods have achieved substantial improvements compared to traditional forms of implementation in that we can now make full use of our cluster, so obtaining substantial improvements compared to traditional techniques. They also have the advantage of being scalable.

## 3 ASSOCIATION RULES

In the data mining and machine learning field, association rules are used to discover facts that often occur together within a particular data set. A typical example of this type of problem is to find out which products in a supermarket are normally purchased together. Different methods for the extraction of association rules have been widely researched and have proved very interesting for discovering relations between the variables in datasets.

Association rules were formally defined for the first time by Agrawal *et al.* [7] as follows.

Let $I = \{i_1, i_2, ..., i_m\}$ be a set of items and $D = \{t_1, t_2, ..., t_n\}$ a set of $n$ transactions in which $t_j$ contains a subset of items. This means that a rule can be defined as follows:

$$X \rightarrow Y,\ where\ X,\ Y \subseteq I\ and\ X \cap Y = \varnothing$$

where $X$ is referred to as the antecedent (or left-hand side of the rule) and $Y$ is the consequent (or right-hand side of the rule).

The problem of discovering association rules is divided into two sub-tasks:

- Finding all the sets above the minimum support threshold, where support is provided by the percentage of transactions in the set.
- These sets are known as frequent sets.
- On the basis of the frequent sets that are found, generate rules which exceed the minimum threshold for confidence or another measurement of interestingness generally given by the user.

We are now going to discuss the most frequently used methods for measuring the importance of the itemsets and the quality of the rules.

### 3.1 Support

Support [8] is a measure of the frequency at which the items of an association rule are found in the data, i.e. the number of transactions in which the items in a rule occur together as a proportion of the total number of transactions. It is normally represented as $Sup_D\ (X)$ in which X is an itemset and D a database. In general, the most interesting association rules are those with a high-support value.

$$Sup_D(X) = \frac{No\ appearances\ in\ \mathrm{D}\ of\ itemset\ X}{Total\ No\ of\ transactions\ in\ D}$$

And the support of a rule will be:

$$Sup_D(X \rightarrow Y) = \frac{No\ appearances\ in\ \mathrm{D}\ of\ itemset\ X \bigcup Y}{Total\ \mathrm{No}\ of\ transactions\ in\ D}$$

### 3.2 Confidence

Given the itemsets X and Y, and the database D, the confidence value [8], represented as $Conf_D(X \rightarrow Y)$, is the conditional probability of Y appearing in those transactions in D that contain X. In other words, confidence refers to cases in which a rule makes a correct prediction and is calculated as follows:

$$Conf_D\left(X \rightarrow Y\right) = \frac{Sup_D(X \bigcup Y)}{Sup_D(X)}$$

### 3.3 Apriori algorithm

In data mining, the Apriori algorithm is used to determine the association rules in a dataset using measures of support and of interestingness, such as confidence. This algorithm is based on previous knowledge (a priori) of the sets of frequent data items. An item is considered frequent if its frequency of appearance in the database is higher than the minimum support threshold.

Agrawal and Srikant [9] identified a fundamental property when they proposed the Apriori algorithm, namely that any subset of a set of frequent items must also be a set of frequent items. With this in mind, the Apriori algorithm obtains the size-1 frequent itemsets and then the size-2, the size-3 and so on until no more frequent itemsets can be found. It then uses the measure of interestingness, e.g. confidence, to determine the set of association rules, using the frequent sets found in the first phase.

## 4 BDARE ALGORITHM

The Apriori algorithm has big problems with large amounts of Big Data as it does multiple scans of the whole database. This means that the execution time increases in line with the number of transactions. In our research, we used Spark to try to improve the Apriori algorithm.

The data were stored in a Big Data architecture, for which we used Hadoop (which allows for replication through its HDFS – Hadoop Distributed File System) and distributed processing using Spark.

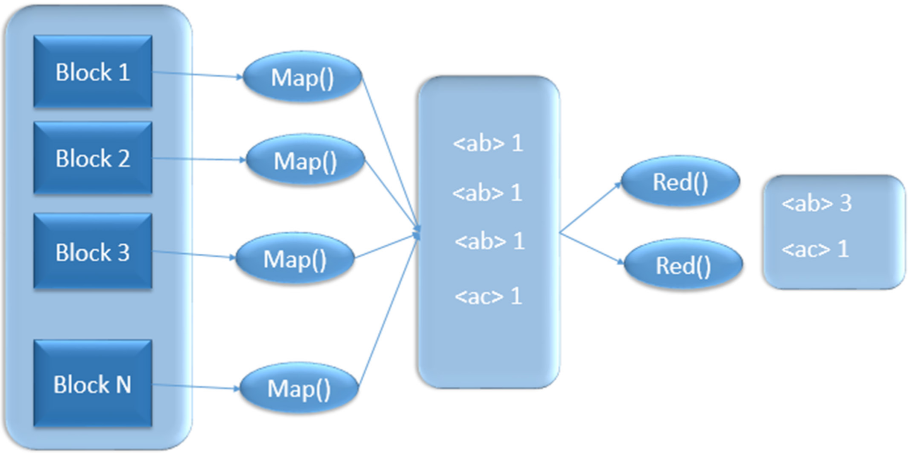


Figure 1: Phase 1 BDARE.

This new algorithm, which we call the BDARE (Big Data Association Rules Extraction) algorithm, is based on two steps. The first involves loading the dataset and calculating how often each item appears in the transactions using the *Map* and *Reduce* functions. The number of times each item appears is counted in two phases. In the *Flatmap()* phase each transaction is transformed into <key, value> pairs with the name of the item as the key and a value of 1, which represents the presence of this item in the transaction <NameItem,1>. A *Reduce()* function then counts the number of appearances of each item.

Figure 1 describes this process, showing the first phase in which the *Flatmap()* function generates the pairs from these transactions and the second, in which the *Reduce()* function returns the number of occurrences of each item in the transactions.

This is followed by step two, in which we consult the size-k-itemsets. For this task, we use a function that returns the candidate k-itemsets from the frequent items. We save these candidates in a dictionary (Python hash table), which we use as a global variable.

After obtaining the size-k candidates we then count these itemsets in the transactions using a process in Spark. The various steps in this process can be seen in Fig. 2. In the top part, you can see the function that calculates the dictionary with the candidate itemsets and at the bottom the process for counting the itemsets.

The algorithm's counting process consists of the following phases:

- **Itemset extraction phase**: In this phase, we use the dictionary of candidate itemsets to extract all the itemsets from each transaction. We use the *FlatMap()* function which when it receives one input returns various outputs, rather than the *Map()* function which for one <key, value> input returns another <returned_key, returned _value> pair.
- **Transformation phase:** In this phase, we have all the itemsets and using the *Map()* function we move from <itemset> to <itemset, 1>.
- **Aggregation phase**: Using a *Reduce()* function, we group each itemset with the sum of the values for each pair.

These phases will be repeated for the calculation of each of the k-itemsets until no new k-itemsets can be calculated.

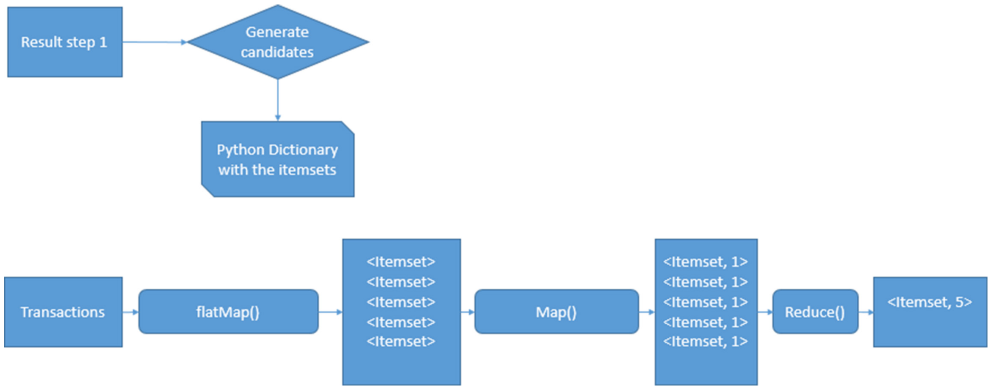


Figure 2: Phase 2 BDARE.

Table 1: Datasets.

| **Name** | **Transactions** | **Items** |
|---|---|---|
| Otto | 63,570 | 1,600 |
| Abalone | 5,000,000 | 88 |

## 5 EXPERIMENTS AND ANALYSIS OF RESULTS

We will now move on to the different experiments. We carried out with the traditional and BDARE algorithms. Our aim was to study the behavior of the algorithm with and without Big Data techniques. To this end, we obtained different datasets with which we could study the behavior of the algorithms with regard to different parameters such as the number of items and the number of transactions. We also studied the execution time and the use of memory in the two algorithms and the results obtained.

In our experiments with algorithms, we used two datasets to assess how well the algorithm functions when the number of transactions or the number of items increases. We can see the different features of dataset in the table 1.

In our experiments, we used as minimum thresholds a support value of 0.2 and a confidence value of 0.8.

The architecture used for the execution of the BDARE algorithm consisted of three machines with Intel Xeon processors with 4 cores at 2.2 GHz, while the traditional algorithm was executed on one of these machines.

### 5.1 Results

We will now look at the performance of the algorithm when the number of transactions or items increases. To this end, we studied the changes in two aspects of the execution of the two algorithms:

- Execution time
- Memory

We began by studying the behavior of the algorithms when the number of transactions increases. To this end, both algorithms were run as subsets of the dataset to observe the behavior with regard to the number of transactions. Figure 3 shows the behavior of the algorithm with the Abalone dataset (the number of transactions has doubled to five million).

In this experiment, we observed that with 1,000 data items the traditional algorithm performed in a similar way to the BDARE algorithm with insignificant differences in execution time. With a small number of data items BDARE does not achieve more efficient times due to the planning of the jobs. However, when the number of transactions increases, the performance of the BDARE algorithm improves. This is because with increasing amounts of data, the planning times become increasingly negligible with respect to the execution time for the algorithm and also because with larger datasets the HDFS makes more partitions, so making better use of the processing capacity. To be exact with the last three datasets (1 million, 2 million and 3 million transactions), we achieved time savings of 8%, 13% and 18%, respectively, compared to the traditional algorithm.

In the graph in Fig. 3, we can also see that with 5 million data items the traditional algorithm did not complete the execution.

Figure 4 shows that when the number of items increases, the algorithm time in BDARE does not offer substantial improvements compared to the traditional algorithm. This is because the Apriori algorithm explores all possible item combinations and in each of these explorations consults the dataset in each transaction.

As can be seen in Fig. 4, the traditional algorithm does not complete its execution due to a fault resulting from lack of memory. This is due to the fact that the Apriori algorithm has to generate all the combinations of items and if these increase, as happens in this experiment, vast memory

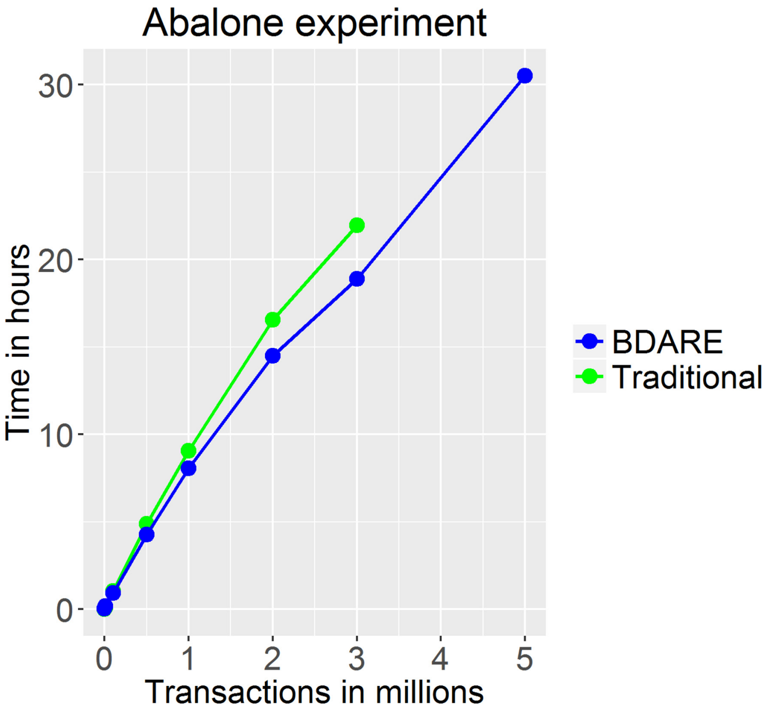


Figure 3: Abalone experiment.

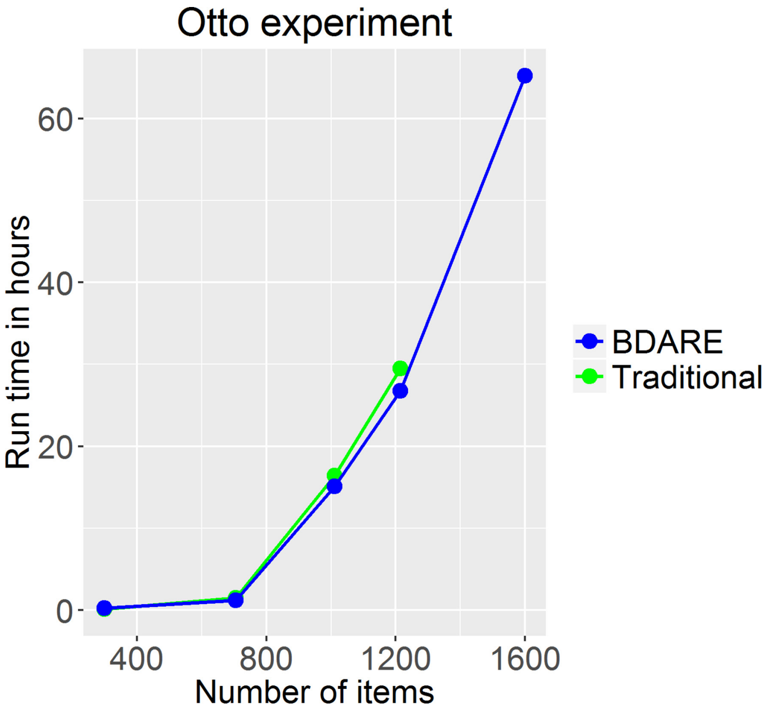


Figure 4: Otto experiment.

capacities are required. Figure 4 also shows that BDARE by contrast does complete the execution with more items thanks to the use of Big Data techniques, which allow us to use the memory of the three nodes, thereby obtaining greater capacity.

## 6 CONCLUSIONS AND FUTURE RESEARCH

As we have seen, this paper has focused on the study of one of the most commonly used techniques in data mining today, association rules, which allow us to extract information from datasets.

The algorithms studied in the bibliography (such as the Apriori algorithm studied here) are not intended for large datasets, as this would involve very high computational costs and decreasing efficiency as the dataset grows.

To this end, we have presented an implementation of this algorithm using Spark, one of the most frequently used Big Data technologies today.

The results we obtained show an improvement in the performance of the algorithm using Spark as compared to the traditional Apriori algorithm. This improvement was not only important in terms of time and processing capacity in memory. An additional advantage is that as our algorithm uses technology such as Hadoop and Spark, performance can easily be improved even further just by expanding our processing system with more nodes. This allows us to scale our algorithms in our own large data centers or in cloud systems such as AWS (Amazon Web Services), another great advantage of Big Data technology.

As regards future research on the results we have obtained, we propose to solve the problems encountered during this project. We will begin by studying algorithms that are more efficient in terms of the number of items such as for example the FP-Grow algorithm [10]. This algorithm outperforms Apriori in that it does not need to make successive consultations of the same transactions.

Association rules are commonly used in a large number of applications. Our BDARE algorithm can be used in some of these applications in which the use of traditional algorithms is not viable due to the vast amount of data that needs to be processed. It could be very useful for example in sensor networks that generate enormous amounts of data in short spaces of time or in social networks in which there are millions of users.

Lastly, we should mention that these procedures can be generalized to other data mining techniques that use association rules such as exceptions and anomalies [4], gradual dependencies [11, 12] etc.

## ACKNOWLEDGEMENT

The research reported in this paper was partially supported by the Andalusian Government (Junta de Andalucía) under project P11-TIC-7460 and by the Energy IN TIME project funded from the European Union under Grant agreement No. 608981

## REFERENCES

[1] Fayyad, Usama M., et al., *Advances in Knowledge Discovery and Data Mining*, 1996. http://dx.doi.org/10.1016/j.ins.2014.03.043

[2] Apache., Hadoop apache, 2015.

[3] Meng, Xiangrui, et al. MLlib: Machine Learning in Apache Spark. *arXiv: preprint arXiv:1505.06807*, 2015

[4] Delgado, M., Ruiz, M.D. & Sánchez, D., New approaches for discovering exception and anomalous rules. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **19**(02), pp. 361–399, 2011. http://dx.doi.org/10.1142/S0218488511007039

[5] del Rio, Sara, et al., On the use of mapreduce for imbalanced big data using random forest. *Information Sciences*, **285**, pp. 112–137, 2014.

[6] Palit, Indranil; Reddy, Chandan K., Scalable and parallel boosting with mapreduce. *Knowledge and Data Engineering, IEEE Transactions on*, 24(10), pp. 1904–1916, 2012.

[7] Agrawal, Rakesh, Imielinski, Tomasz & Swami, Arun, Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, **22**, pp. 207–216, 1993. http://dx.doi.org/10.1145/170036.170072

[8] Berzal, Fernando, et al., An alternative approach to discover gradual dependencies. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **15**(05), pp. 559–570, 2007. http://dx.doi.org/10.1142/S021848850700487X

[9] Agrawal, Rakesh, et al., Fast algorithms for mining association rules. *In Proceedings of 20th international Conference Very Large Data Bases, VLDB*, pp. 487–499, 1994.

[10]Jiawei, H., Jian, P. & Yiwen, Y., Mining frequent patterns without candidate generation. *ACM SIGMOD Record*, 29, pp. 1–12, 2000. http://dx.doi.org/10.1145/335191.335372

[11]Berzal, Fernando, et al., A new framework to assess association rules. *In Advances in Intelligent Data Analysis*, Springer Berlin: Heidelberg, pp. 95–104, 2001. http://dx.doi.org/10.1007/3-540-44816-0_10

[12] Hüllermeier, E., Association rules for expressing gradual dependencies. *In Principles of Data Mining and Knowledge Discovery*, Springer Berlin: Heidelberg, pp. 200–211, 2002. http://dx.doi.org/10.1007/3-540-45681-3_17

## 1.2 Spark solutions for distributed association rules mining extraction algorithms (Submitted to Pattern recognition letters)

- Carlos Fernandez Basso, M. Dolores Ruiz, Maria J. Martin-Bautista.
  - Status: **Submitted to Pattern recognition letters**.
  - Impact Factor (JCR 2019): **3.73**
  - Subject Category: **Computer Science, Information System**
  - Rank: **50/134**
  - Quartile: **Q2**

# Fuzzy Association Rules Mining Using Spark

Carlos Fernandez-Bassso[1], M. Dolores Ruiz[2]( ), and Maria J. Martin-Bautista[1]

[1] Computer Science and A.I. Department, CITIC-UGR, University of Granada, Granada, Spain
{cjferba,mbautis}@decsai.ugr.es

[2] Computer Engineering Department, University of Cádiz, Cádiz, Spain
mariadolores.ruiz@uca.es

**Abstract.** Discovering new trends and co-occurrences in massive data is a key step when analysing social media, data coming from sensors, etc. Traditional Data Mining techniques are not able, in many occasions, to handle such amount of data. For this reason, some approaches have arisen in the last decade to develop parallel and distributed versions of previously known techniques. Frequent itemset mining is not an exception and in the literature there exist several proposals using not only parallel approximations but also Spark and Hadoop developments following the MapReduce philosophy of Big Data.

When processing fuzzy data sets or extracting fuzzy associations from crisp data the implementation of such Big Data solutions becomes crucial, since available algorithms increase their execution time and memory consumption due to the problem of not having Boolean items. In this paper, we first review existing parallel and distributed algorithms for frequent itemset and association rule mining in the crisp and fuzzy case, and afterwards we develop a preliminary proposal for mining not only frequent fuzzy itemsets but also fuzzy association rules. We also study the performance of the proposed algorithm in several datasets that have been conveniently fuzzyfied obtaining promising results.

**Keywords:** Big data algorithms · Fuzzy frequent itemset
Fuzzy association rules · Data Mining · Apriori

## 1 Introduction

The vast amounts of data generated, stored and analysed by companies, and by extension by private users, has given rise to a new phenomenon known as Big Data. Every day millions of tweets are published on Twitter, countless messages are sent via messaging apps and multitudes of comments are generated in online shops. In addition to this, every year more buildings are summed to the "smart sensored" fashion to collect data and use it in their daily performance to be more efficient. The necessity of constantly extracting information from all the gathered

J. Medina et al. (Eds.): IPMU 2018, CCIS 854, pp. 15–25, 2018.
https://doi.org/10.1007/978-3-319-91476-3_2

data is a fact, and the Big Data philosophy using MapReduce framework enables it. In particular, Data Mining techniques are currently under development to benefit from this new framework [3,5,11,12].

One important technique, often employed for exploratory analysis, is that of association rules. They have the form of implications $A \rightarrow B$, which represent the joint co-occurrence of A and B. However, many of the data to be analysed have a nature that is difficult to represent, such as natural language texts. Beside this, when discovering associations between numerical variables we have to take care when data is discretized since final results can vary a lot depending on how the ranges are defined [17]. To better represent this kind of data, Fuzzy Sets theory [26] has proved to be a good option, having as a result fuzzy databases where we can search for fuzzy association rules [6].

In this paper we propose a solution to perform this analysis. There are various methods for mining fuzzy association rules which enable us to analyse and extract interesting information from datasets. These methods run into problems when they are used to analyse vast amounts of data, becoming less efficient at processing and analysis. To this end, we propose a new technique for mining fuzzy association rules that enables the processing of big amounts of data. We have implemented it using Spark [8] which enable faster memory operations than Hadoop [25] since it allows in-memory computations. The results of our experiments show that this method improves the efficiency of the algorithm, with respect to traditional techniques, in terms of time and memory when the number of transactions increases. However when the number of items increases, the algorithm does not offer in all cases significant efficiency improvements in terms of execution time compared to traditional methods. Nonetheless, thanks to the fact that the Big Data techniques offer greater memory capacity, substantial improvements can be achieved when memory problems arise in the generation of the item combinations to be analysed in massive datasets.

The paper is organized as follows: Sect. 2 reviews the literature to show how Big Data technologies can improve existent Data Mining algorithms. Section 3 introduces the measures and methods employed to mine fuzzy association rules. Next section presents the BDFARE algorithm developed for mining fuzzy association rules employing Big Data technologies. Section 5 shows the experiments and results obtained, prior to concluding the paper in Sect. 6.

## 2 Preliminary Concepts and Related Work

In the literature there are several approaches for mining frequent itemsets using Big Data techniques. The most famous framework, called *MapReduce* was designed by Google in 2003. Since then, there have been proposed several ways to perform association rule analysis with some minor changes. The MapReduce framework bases in two different functions to distribute the computation. On one hand, the *Map()* function transforms data into $(key, value)$ pairs according to some criteria, and on the other hand, the *Reduce()* function aggregates the lists of key-value pairs sharing the same key to obtain a piece of processed data.

There are two different frameworks for distributed processing of data: Hadoop and Spark. Some of the methods that are already implemented in these platforms include RandomForest and K-means (clustering) within the official Spark MLlib library [8]. These methods have achieved substantial improvements compared to traditional forms of implementation in the sense that we can now make full use of our cluster, obtaining thus substantial improvements compared to traditional techniques and more scalable algorithms. In particular, in Spark it is included the PFP (Parallel FP-Growth) which is a distributed version of FP-Growth to obtain the frequent itemsets of higher level exceeding the minimum support threshold [18].

In addition to this, we can find in the literature other proposals for mining frequent itemsets using MapReduce techniques for the non-fuzzy case. We can highlight some approaches implementing Apriori extensions using hadoop: [9,10,19]. As mentioned, Spark accelerates the performance versus Hadoop implementations since it makes computations in memory. In addition to this, the algorithms proposed in [9,10,19] search directly in the data the itemsets instead of using other data structures, e.g. trees, hash tries or hash tables, which can decrease time execution as concluded in a study made in [24]. Spark frameworks for Apriori extensions can be found in [21,22]. The R-Apriori and YAFIM algorithms proposed in [21,22] respectively, are very similar to the non-fuzzy phase for each $\alpha$-cut of our approach but they make a loop to search k-itemsets inside the distributed process using a hash tree while we make the MapReduce for every k-itemset using a hash table.

To the best of our knowledge there is only one work presenting how to discover fuzzy association rules employing the MapReduce framework. This work [14] is based on an extension to the fuzzy case of the Count Distribution algorithm [13,15].

## 3 Fuzzy Association Rules

Association rules were formally defined for the first time by Agrawal et al. [1]. The problem consists in discovering implications of the form $X \rightarrow Y$ where $X, Y$ are subsets of items from $I = i_1, i_2, ..., i_m$ fulfilling that $X \cap Y = \emptyset$ in a database formed by a set of $n$ transactions $D = t_1, t_2, ..., t_n$ each of them containing subsets of items from $I$. $X$ is usually referred as the antecedent and $Y$ as the consequent of the rule.

The problem of discovering association rules is divided into two sub-tasks:

- Finding all the sets above the minimum support threshold, where support is provided by the percentage of transactions in the set. These sets are known as frequent sets.
- On the basis of the frequent sets are found, rules are discovered as those exceeding the minimum threshold for confidence or another measurement of interestingness generally given by the user.

However, the nature of the data can be diverse and can come described in numerical, categorical, imprecisely, etc. In the case of numerical elements, a first approximation could be to categorise them so that, for example, the height of a person may be given by a range to which it belongs, as for instance [1.70, 1.90]. However, depending on how these intervals are defined, the obtained results may vary a lot. To avoid this, the use of linguistic labels such as "high" represented by a fuzzy set is a good option to represent the height of a person having at the same time a meaningful semantic to the user. Beside this, we may also have a dataset with imprecise knowledge where ordinary crisp methods cannot be directly applied.

To deal with this kind of data we introduce the concept of fuzzy transaction and fuzzy association rule defined in [4,6].

**Definition 1.** *Let $I$ be a set of items. A fuzzy transaction, $t$, is a non-empty fuzzy subset of $I$ in which the membership degree of an item $i \in I$ in $t$ is represented by a number in the range [0, 1] and denoted by $t(i)$.*

By this definition a crisp transaction is a special case of fuzzy transaction. We denote by $\tilde{D}$ a database consisting in a set of fuzzy transactions. For an itemset, $A \subseteq I$, the degree of membership in a fuzzy transaction $t$ is calculated as the minimum of the membership degree of all its items:

$$t(A) = \min_{i \in A} t(i). \tag{1}$$

Then, a fuzzy association rule $A \rightarrow C$ is satisfied in $\tilde{D}$ if and only if $t(A) \leq t(C)\ \forall t \in \tilde{D}$, that is, the degree of satisfiability of $C$ in $\tilde{D}$ is greater than or equal to the degree of satisfiability of $A$ for all fuzzy transactions $t$ in in $\tilde{D}$.

Using this model the support and confidence measures are defined using a semantic approach based on the evaluation of quantified sentences as proposed in [4,6]. Using the $GD$-method [6] and the quantifier $Q_M(x) = x$ the support of a fuzzy rule $A \rightarrow B$ results:

$$FSupp(A \rightarrow B) = \sum_{\alpha_i \in \Lambda(A \cap B)} (\alpha_i - \alpha_{i-1}) \frac{|(A \cap B)_{\alpha_i}|}{|\tilde{D}|} \tag{2}$$

where $\Lambda(A \cap B) = \{\alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of $\alpha$-cuts. In the previous formula, by abuse of notation, we consider the associated fuzzy set to a set of items, i.e. $X_{\alpha_i}$ represents the $\alpha$-cut of the fuzzy set derived from the itemset $X$, which is the fuzzy set with membership degree $\mu_X(t) = t(X) = \min_{i \in X} t(i)$ where $X \subset I$. Note that the elements of the fuzzy set derived from $X$ are the transactions.

Analogously the confidence is computed as follows:

$$FConf(A \rightarrow B) = \sum_{\alpha_i \in \Lambda(A \cap B)} (\alpha_i - \alpha_{i-1}) \frac{|(A \cap B)_{\alpha_i}|}{|A_{\alpha_i}|} \tag{3}$$

Employing support and confidence measures and setting appropriated thresholds for them fuzzy association rules can be discovered. In [7] it is proposed a parallelization of the computation of $FSupp$ and $FConf$ using a set of predefined $\alpha$-cuts. Note that considering a sufficiently dense set of $\alpha$-cuts in the unit interval, the obtained measure will be a good approximation of the real measure that should consider every $\alpha \in [0, 1]$ appearing in the dataset. This is the main idea of our proposal for mining fuzzy association rules using MapReduce. Firstly, the algorithm developed using MapReduce is applied repeatedly for every $\alpha$-cut. The final step consists in applying a MapReduce phase which aggregates the results using the previous formulas for support and confidence.

## 4 BDFARE Algorithm

Traditional algorithms have some problems when dealing with large amounts of data as they make multiple scans of the whole database. This means that the execution time increases with the number of transactions. In our research we have used Spark to improve Apriori algorithm for mining fuzzy association rules as follows.

The data is stored in a Big Data architecture, for which we used Hadoop (which allows for replication through its HDFS - Hadoop Distributed File System) and enables distributed processing using Spark.

This new algorithm, called BDFARE (Big Data Fuzzy Association Rules Extraction) algorithm, is based on two phases. The first one involves loading the dataset and calculating how often each item appears in the set of transactions using the Map and Reduce functions. In Fig. 1 we can see an example of first phase with the value of $\alpha = 0.5$. In this example, the $Map()$ function, in particular it is used the $FlatMap()$ function, returns only the items with support higher than 0.5 (column in the middle only contains items with membership values $\geq 0.5$). After that, a re-counting of the obtained items is done using the $Reduce()$ function.

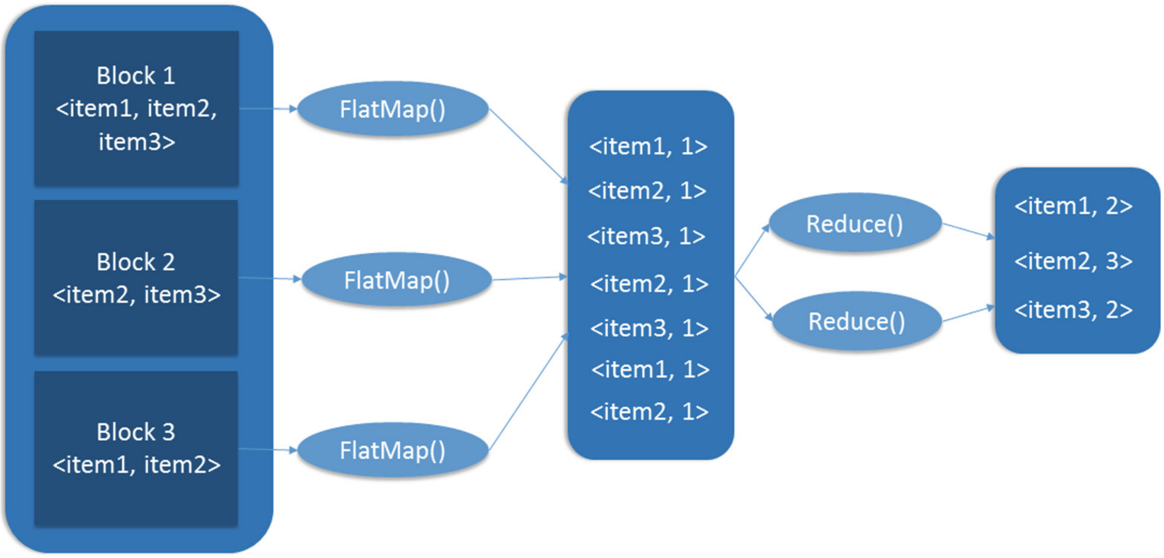


**Fig. 1.** First phase of BDFARE

This is followed by the second phase, in which we extract the size-$k$ itemsets (see Fig. 2). For this task we used a function that returns the candidate $k$-itemsets from the frequent items. As a result, a list of pairs of the type $<itemset, degree\_of\_membership>$ is saved in a global variable representing the candidate list. This enables a faster access to the list. After this, the $map()$ function returns, as in the previous phase, the counts of the candidates, having a pair $<itemset, 1>$ when the membership value of the itemset is higher than the value of the $\alpha$ considered in that iteration. In this way, the algorithm calculates in each step all the cardinalities necessary for the computation of the final support and confidence measures. These two phases will be repeated until no new $k$-itemsets can be found.

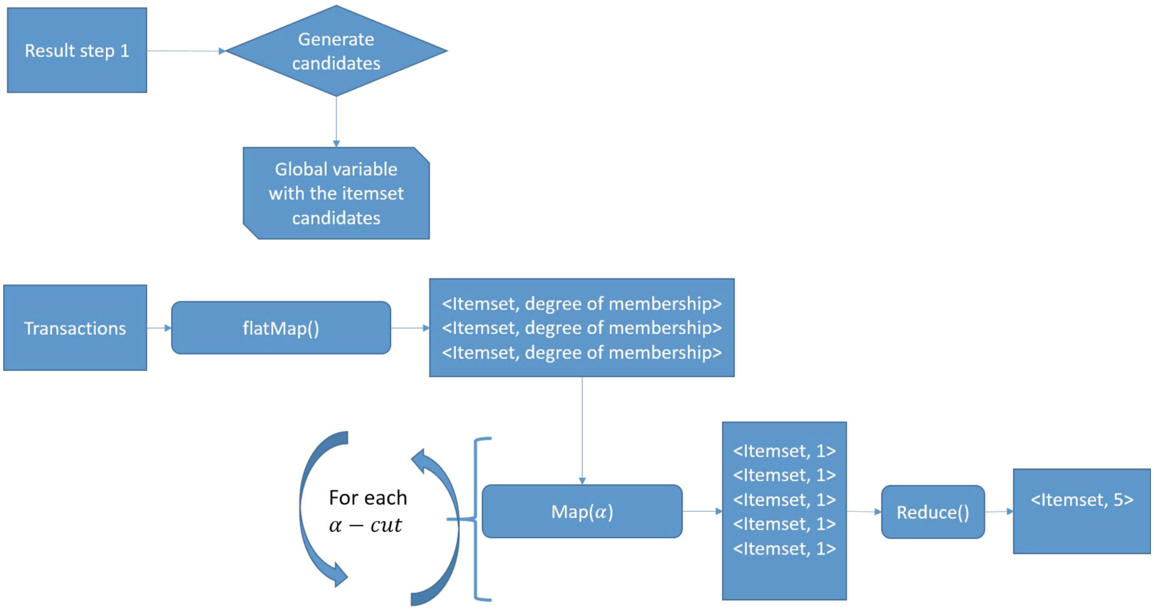


**Fig. 2.** Second phase of BDFARE

As mentioned in Sect. 2, we have employed a hash table as the central data structure to accelerate the searching of itemsets. If we use instead of the hash table a linear search at each node it will result in an increase of time. In [24] a comparison between the different data structures was made concluding that the hash table outperformed among the three data structures (hash tree, trie and hash table) for both real-life and synthetic datasets.

## 5 Experiments and Results

In order to check the performance of our proposal we have carried out several experiments to compare running time using the non-distributed version of Apriori for discovering fuzzy association rules with the distributed proposal using BDFARE algorithm. Our aim is to study the behaviour of the algorithm with and without Big Data techniques. To this end, we applied the algorithms in several fuzzy transactional datasets and we study their running time according to different parameters: the number of items and the number of transactions of the datasets.

Three different datasets have been considered from the UCI machine learning repository[1] where some attributes have been conveniently fuzzyfied as described in [23]. The German dataset consists of transactions about credits offered by a german bank. Three variables were fuzzyfied: amount of the credit, its duration and the age of the person who owns the credit. The Autompg dataset consists of several attributes about cars. In this case, the continuous attributes were fuzzified using the following linguistic labels: low, medium and high. The Bank dataset contains data about marketing campaigns of a Portuguese banking institution. In this dataset we have fuzzified their continuous attributes by defining a suitable fuzzy partition according to the semantics of the attribute (description of the fuzzy sets employed can be found in [23]).

The final datasets used in the experiments have been replicated in order to obtain larger datasets to prove the performance of the algorithm in extreme situations. Their original size can be found in Table 1. Actually, there is not any large fuzzy dataset available in open data repositories, but we plan to apply the algorithm to data collected during a time period from sensored buildings. These sensors give numerical values from a continuous scale that are very close among them (e.g. 25° and 25.2°). In this case, the building operators are more interested in obtaining patterns relating temperatures such as “warm”, “cold”, “very cold”, etc. that can be represented by convenient fuzzy sets instead of using intervals that may divide very close data in two different intervals.

**Table 1.** Datasets

| Fuzzy database | Transactions | Items |
|---|---|---|
| German | 1000 | 79 |
| Autompg | 398 | 39 |
| Bank | 45211 | 112 |

### 5.1 Results

The experimental evaluation have been made in an computer architecture consisting of a cluster comprised of three processing units with Intel Xeon processors with 4 cores at 2.2 GHz, while the traditional algorithm was executed on one of these clusters. We have performed several experiments with different thresholds values, but we show here the results for minimum support equal to 0.2 and minimum confidence equal to 0.8. The important thing here is to set the same thresholds for distributed and non-distributed approaches since we are

[1] http://archive.ics.uci.edu/ml/.

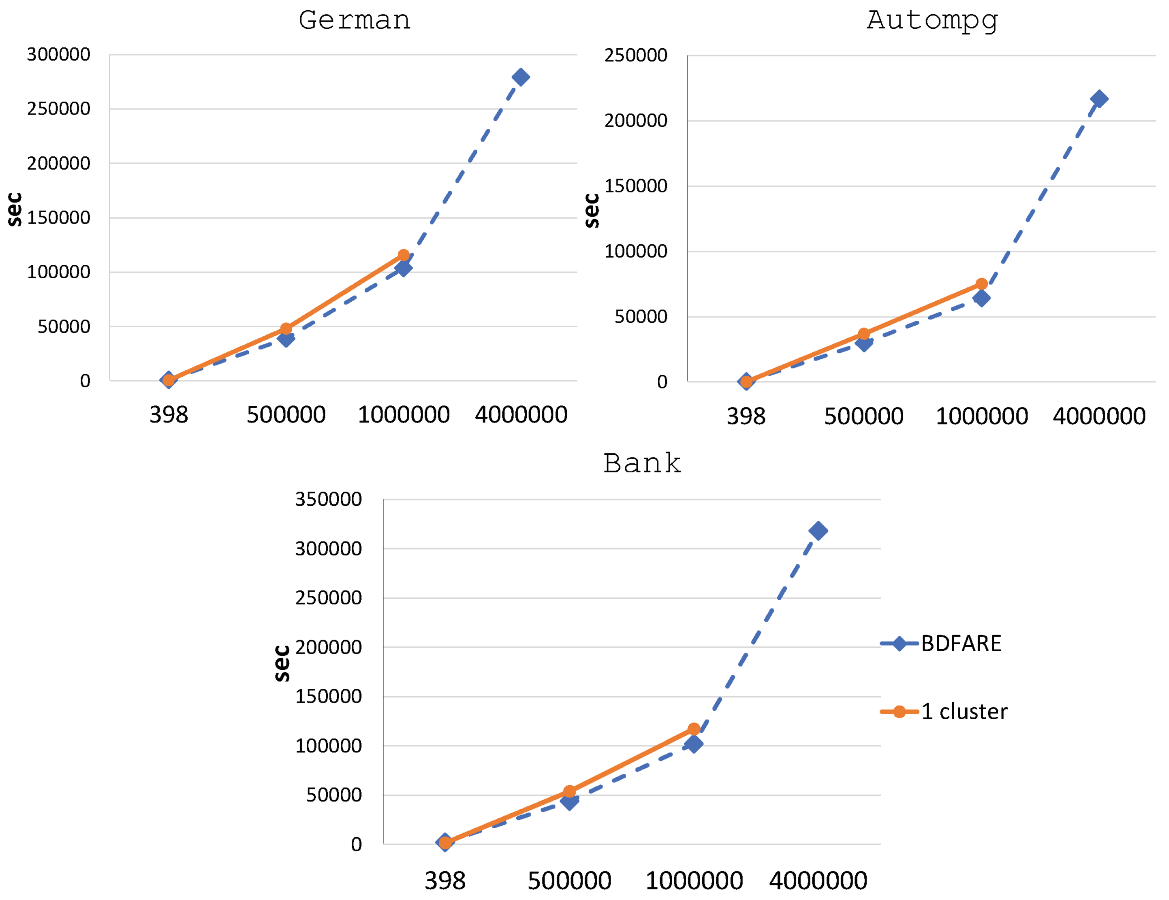


**Fig. 3.** Performance of BDFARE vs non-distributed algorithm when the quantity of transactions increases

more interested here in observing the performance (time and memory) of both approaches depending on the number of transactions and the number of items.

We began by studying the behaviour of the algorithms (distributed and non-distributed) when the number of transactions increases. To this end, they were run as subsets of the whole datasets in order to observe the behaviour with regard to the number of transactions. Figure 3 shows the behaviour of the algorithm when the quantity of transactions increases in each of the datasets (the number of transactions has been replicated till obtaining four millions). It can be observed that the traditional algorithm performed worst than the BDFARE algorithm, as expected.

To be exhaustive, with the datasets consisted of 0.5 million and 1 million transactions, we achieved time savings in average of 12% and 18% respectively compared to the non-distributed version. We can also see in this graph that the traditional algorithm did not complete its execution for the dataset containing 4 million transactions due to an error resulting from the lack of memory. By contrast, BDFARE completed the execution thanks to the use of Big Data techniques, which allow us to use the memory of the three nodes, thereby obtaining a higher capacity.

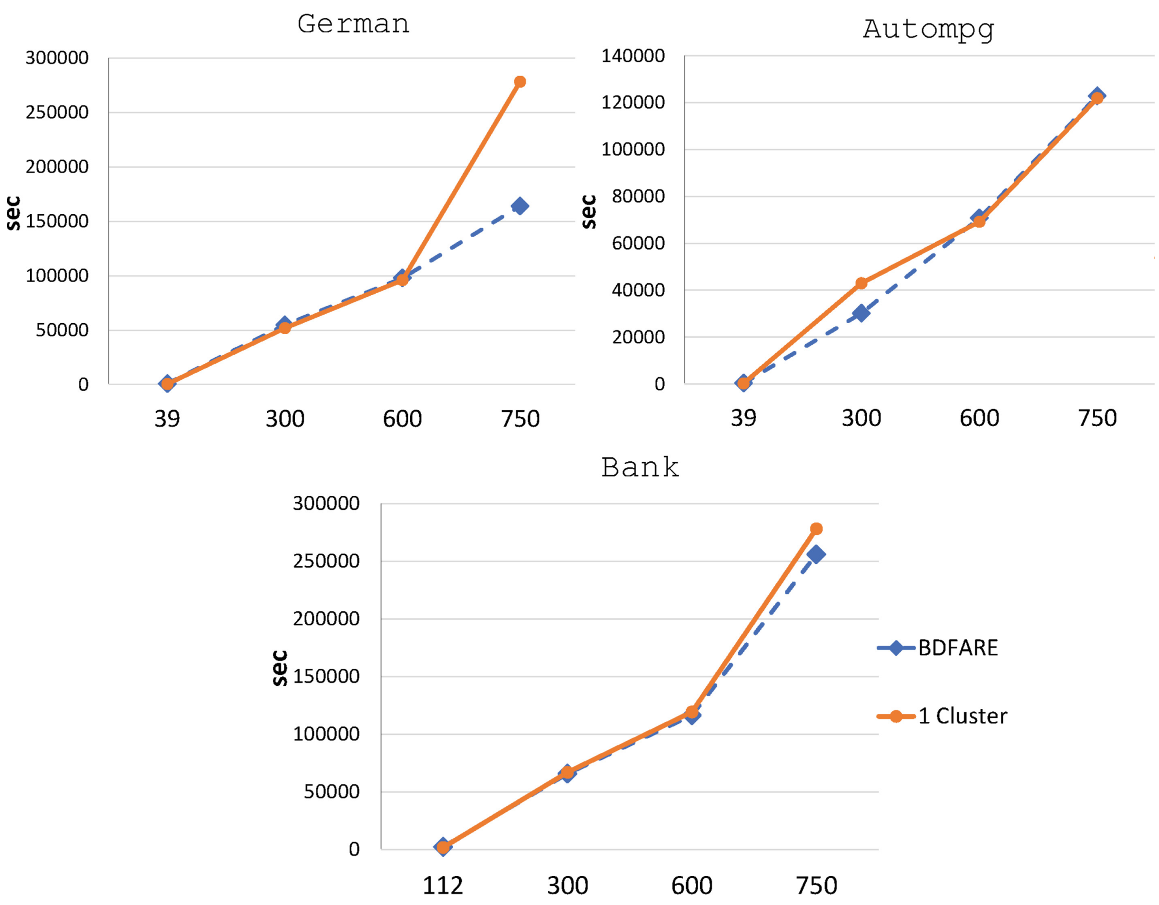


**Fig. 4.** Performance of BDFARE vs non-distributed algorithm when the quantity of items increases

Figure 4 shows the running time when the number of items increases in the three datasets. In this graph we observe that the execution time using BDFARE does not offer substantial improvements in some cases compared to the traditional algorithm. This is because the Apriori algorithm explores all possible item combinations and in each of these explorations it consults the dataset. Additionally, with a small number of items BDFARE does not achieve always more efficient executions due to the time consumed in the planning of the jobs, necessary when distributing data. However when the number of transactions increases, the performance of the BDFARE algorithm tends to improve the non-distributed approach.

## 6 Conclusions and Future Research

As we have seen, this paper has focused on the study of one of the most commonly used techniques in data mining, association rules, which allows to extract co-occurrence patterns from datasets. The algorithms proposed traditionally for mining association rules fail when analysing massive datasets because the process results in very high computational costs and its efficiency decreases when the dataset grows.

To this end we have presented an extension to Apriori algorithm for mining fuzzy association rules using Spark, a Big Data framework which enables MapReduce implementations. The algorithm has been compared with non-distributed version of the algorithm showing improvements not only in terms of execution time and but also in terms of memory, improving the processing capacity, since some of the experiments were not able to process with the non-distributed version. An additional advantage is that our algorithm and its performance can be easily improved even further just by expanding our processing system with more clusters (computation nodes). This allows to scale our approach in external data centers or in cloud systems such as AWS (Amazon Web Services), another great advantage of Big Data technology.

As regards future research, we want to implement more efficient approaches [16] that have been proved that performs quite well in the non-distributed case such as the Apriori-TID [2], FP-Growth [20] or ECLAT [27] algorithms. Additionally, we plan to apply the presented approach and the new implementations to sensor data collected from several buildings in order to study the efficiency patterns relating indoor and outdoor temperatures, HVAC (Heating, ventilation, air-conditioning) set points and energy consumptions. In this case, fuzzy sets are suitable to represent understandable value ranges for the users, building operators, etc.

**Acknowledgements.** The research reported in this paper was partially supported by the Andalusian Government (Junta de Andalucía) under projects P11-TIC-7460 and the Spanish Ministry for Economy and Competitiveness by the project grant TIN2015-64776-C3-1-R.

## References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining associations between sets of items in large databases. In: ACM-SIGMOD International Conference on Data, pp. 207–216 (1993)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the Twentieth International Conference on Very Large Databases, Santiago, Chile, pp. 487–499 (1994)
3. Anastasiu, D.C., Iverson, J., Smith, S., Karypis, G.: Big data frequent pattern mining. In: Aggarwal, C.C., Han, J. (eds.) Frequent Pattern Mining, pp. 225–259. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07821-2_10
4. Berzal, F., Delgado, M., Sánchez, D., Vila, M.A.: Measuring accuracy and interest of association rules: a new framework. Intell. Data Anal. **6**(3), 221–235 (2002)
5. del Río, S., López, V., Benítez, J.M., Herrera, F.: On the use of MapReduce for imbalanced big data using random forest. Inf. Sci. **285**, 112–137 (2014). Processing and Mining Complex Data Streams
6. Delgado, M., Marín, N., Sánchez, D., Vila, M.A.: Fuzzy association rules: general model and applications. IEEE Trans. Fuzzy Syst. **11**(2), 214–225 (2003)
7. Delgado, M., Ruiz, M.D., Sánchez, D., Serrano, J.M.: A formal model for mining fuzzy rules using the RL representation theory. Inf. Sci. **181**(23), 5194–5213 (2011)
8. Meng, X., et al.: MLlib: machine learning in apache spark. arXiv preprint: abs/1505.06807 (2015)

9. Farzanyar, Z., Cercone, N.: Accelerating frequent itemset mining on the cloud: a MapReduce-based approach. In: IEEE 13th International Conference on Data Mining Workshops, pp. 592–598 (2013)
10. Farzanyar, Z., Cercone, N.: Efficient mining of frequent itemsets in social network data based on MapReduce framework. In: Proceedings of the 2013 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013), pp. 1183–1188 (2013)
11. Fernández, A., Carmona, C.J., del Jesus, M.J., Herrera, F.: A view on fuzzy systems for big data: progress and opportunities. Int. J. Comput. Intell. Syst. **9**, 69–80 (2016)
12. Fernandez-Basso, C., Ruiz, M.D., Martin-Bautista, M.J.: Extraction of association rules using big data technologies. Int. J. Des. Nat. Ecodyn. **11**(3), 178–185 (2016)
13. Gabroveanu, M., Cosulschi, M., Constantinescu, N.: A new approach to mining fuzzy association rules from distributed databases. Ann. Univ. Bucharest **54**, 3–16 (2005)
14. Gabroveanu, M., Cosulschi, M., Slabu, F.: Mining fuzzy association rules using MapReduce technique. In: International Symposium on INnovations in Intelligent SysTems and Applications, INISTA, pp. 1–8 (2016)
15. Gabroveanu, M., Iancu, I., Cosulschi, M., Constantinescu, N.: Towards using grid services for mining fuzzy association rules. In: Proceedings of the 1st East European Workshop on Rule-Based Applications, RuleApps, pp. 507–513 (2007)
16. Hipp, J., Güntzer, U., Nakhaeizadeh, G.: Algorithms for association rule mining - a general survey and comparison. ACM SIGKDD Explor. Newsl. **2**(1), 58–64 (2000)
17. Hüllermeier, E., Yi, Y.: In defense of fuzzy association analysis. IEEE Trans. Syst. Man Cybern. Part B Cybern. **37**(4), 1039–1043 (2007)
18. Li, H., Wang, Y., Zhang, D., Zhang, M., Chang, E.Y.: PFP: parallel FP-growth for query recommendation. In: Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 107–114. ACM (2008)
19. Li, N., Zeng, L., He, Q., Shi, Z.: Parallel implementation of Apriori algorithm based on MapReduce. In: Proceedings of the 2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2012, pp. 236–241. IEEE Computer Society, Washington, D.C. (2012)
20. Pei, J., Yin, Y., Mao, R., Han, J.: Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Mining Knowl. Discov. **8**(1), 53–87 (2004)
21. Qiu, H., Gu, R., Yuan, C., Huang, Y.: YAFIM: a parallel frequent itemset mining algorithm with spark. In: 2014 IEEE International Parallel & Distributed Processing Symposium Workshops (IPDPSW), pp. 1664–1671. IEEE (2014)
22. Rathee, S., Kaul, M., Kashyap, A.: R-Apriori: an efficient Apriori based algorithm on spark. In: Proceedings of the PIKM 2015, Melbourne, VIC, Australia. ACM (2015)
23. Ruiz, M.D., Sánchez, D., Delgado, M., Martin-Bautista, M.J.: Discovering fuzzy exception and anomalous rules. IEEE Trans. Fuzzy Syst. **24**(4), 930–944 (2016)
24. Singh, S., Garg, R., Mishra, P.K.: Performance analysis of Apriori algorithm with different data structures on hadoop cluster. Int. J. Comput. Appl. **128**(9), 45–51 (2015)
25. White, T.: Hadoop: The Definitive Guide, 4th edn. O'Reilly, Sebastopol (2015)
26. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
27. Zaki, M.J.: Scalable algorithms for association mining. IEEE Trans. Knowl. Data Eng. **12**(3), 372–390 (2000)

# 2 Fuzzy association rules mining in Big Data

## 2.1 Fuzzy association rules mining using Spark (IPMU congress)

- Carlos Fernandez-Basso, M. Dolores Ruiz, Maria J. Martin-Bautista. IPMU'2018.
  - Status: **Published**.
  - 17th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems. Cádiz, Spain, June 11th – 15th (2018)

# Fuzzy association rules mining using Spark

Carlos Fernandez-Bassso[1], M. Dolores Ruiz[2], and Maria J. Martin-Bautista[1]

[1] Computer Science and A.I. Department, CITIC-UGR,
University of Granada, Spain
[2] Computer Engineering Department, University of Cádiz, Spain
{cjferba,mbautis}@decsai.ugr.es,mariadolores.ruiz@uca.es

**Abstract.** Discovering new trends and co-occurrences in massive data is a key step when analysing social media, data coming from sensors, etc. Traditional Data Mining techniques are not able, in many occasions, to handle such amount of data. For this reason, some approaches have arisen in the last decade to develop parallel and distributed versions of previously known techniques. Frequent itemset mining is not an exception and in the literature there exist several proposals using not only parallel approximations but also Spark and Hadoop developments following the MapReduce philosophy of Big Data.

When processing fuzzy data sets or extracting fuzzy associations from crisp data the implementation of such Big Data solutions becomes crucial, since available algorithms increase their execution time and memory consumption due to the problem of not having Boolean items. In this paper, we first review existing parallel and distributed algorithms for frequent itemset and association rule mining in the crisp and fuzzy case, and afterwards we develop a preliminary proposal for mining not only frequent fuzzy itemsets but also fuzzy association rules. We also study the performance of the proposed algorithm in several datasets that have been conveniently fuzzyfied obtaining promising results.

**Keywords:** Big Data algorithms, Fuzzy frequent itemset, Fuzzy Association Rules, Data Mining, Apriori

## 1 Introduction

The vast amounts of data generated, stored and analysed by companies, and by extension by private users, has given rise to a new phenomenon known as Big Data. Every day millions of tweets are published on Twitter, countless messages are sent via messaging apps and multitudes of comments are generated in online shops. In addition to this, every year more buildings are summed to the "smart sensored" fashion to collect data and use it in their daily performance to be more efficient. The necessity of constantly extracting information from all the gathered data is a fact, and the Big Data philosophy using MapReduce framework enables it. In particular, Data Mining techniques are currently under development to benefit from this new framework [5, 3, 11, 12].

One important technique, often employed for exploratory analysis, is that of association rules. They have the form of implications $A \to B$, which represent the

joint co-occurrence of A and B. However, many of the data to be analysed have a nature that is difficult to represent, such as natural language texts. Beside this, when discovering associations between numerical variables we have to take care when data is discretized since final results can vary a lot depending on how the ranges are defined [17]. To better represent this kind of data, Fuzzy Sets theory [26] has proved to be a good option, having as a result fuzzy databases where we can search for fuzzy association rules [6].
In this paper we propose a solution to perform this analysis. There are various methods for mining fuzzy association rules which enable us to analyse and extract interesting information from datasets. These methods run into problems when they are used to analyse vast amounts of data, becoming less efficient at processing and analysis. To this end, we propose a new technique for mining fuzzy association rules that enables the processing of big amounts of data. We have implemented it using Spark [8] which enable faster memory operations than Hadoop [25] since it allows in-memory computations. The results of our experiments show that this method improves the efficiency of the algorithm, with respect to traditional techniques, in terms of time and memory when the number of transactions increases. However when the number of items increases, the algorithm does not offer in all cases significant efficiency improvements in terms of execution time compared to traditional methods. Nonetheless, thanks to the fact that the Big Data techniques offer greater memory capacity, substantial improvements can be achieved when memory problems arise in the generation of the item combinations to be analysed in massive datasets.
The paper is organized as follows: Section 2 reviews the literature to show how Big Data technologies can improve existent Data Mining algorithms. Section 3 introduces the measures and methods employed to mine fuzzy association rules. Next section presents the BDFARE algorithm developed for mining fuzzy association rules employing Big Data technologies. Section 5 shows the experiments and results obtained, prior to concluding the paper in Section 6.

## 2 Preliminary concepts and related work

In the literature there are several approaches for mining frequent itemsets using Big Data techniques. The most famous framework, called *MapReduce* was designed by Google in 2003. Since then, there have been proposed several ways to perform association rule analysis with some minor changes. The MapReduce framework bases in two different functions to distribute the computation. On one hand, the *Map()* function transforms data into $(key, value)$ pairs according to some criteria, and on the other hand, the *Reduce()* function aggregates the lists of key-value pairs sharing the same key to obtain a piece of processed data.

There are two different frameworks for distributed processing of data: Hadoop and Spark. Some of the methods that are already implemented in these platforms include RandomForest and K-means (clustering) within the official Spark MLlib library [8]. These methods have achieved substantial improvements compared to traditional forms of implementation in the sense that we can now make full use of our cluster, obtaining thus substantial improvements compared to traditional techniques and more scalable algorithms. In particular, in Spark it is included the PFP (Parallel FP-Growth) which is a distributed version of FP-Growth to obtain the frequent itemsets of higher level exceeding the minimum support threshold [18].

In addition to this, we can find in the literature other proposals for mining frequent itemsets using MapReduce techniques for the non-fuzzy case. We can highlight some approaches implementing Apriori extensions using hadoop: [19, 10, 9]. As mentioned, Spark accelerates the performance versus Hadoop implementations since it makes computations in memory. In addition to this, the algorithms proposed in [19, 10, 9] search directly in the data the itemsets instead of using other data structures, e.g. trees, hash tries or hash tables, which can decrease time execution as concluded in a study made in [24]. Spark frameworks for Apriori extensions can be found in [22] and [21]. The R-Apriori and YAFIM algorithms proposed in [22] and [21] respectively, are very similar to the non-fuzzy phase for each $\alpha$-cut of our approach but they make a loop to search k-itemsets inside the distributed process using a hash tree while we make the MapReduce for every k-itemset using a hash table.

To the best of our knowledge there is only one work presenting how to discover fuzzy association rules employing the MapReduce framework. This work [14] is based on an extension to the fuzzy case of the Count Distribution algorithm [15, 13].

## 3 Fuzzy Association Rules

Association rules were formally defined for the first time by Agrawal et al. [1]. The problem consists in discovering implications of the form $X \to Y$ where $X, Y$ are subsets of items from $I = i_1, i_2, ..., i_m$ fulfilling that $X \cap Y = \emptyset$ in a database formed by a set of $n$ transactions $D = t_1, t_2, ..., t_n$ each of them containing subsets of items from $I$. $X$ is usually referred as the antecedent and $Y$ as the consequent of the rule.

The problem of discovering association rules is divided into two sub-tasks:

- Finding all the sets above the minimum support threshold, where support is provided by the percentage of transactions in the set. These sets are known as frequent sets.
- On the basis of the frequent sets are found, rules are discovered as those exceeding the minimum threshold for confidence or another measurement of interestingness generally given by the user.

However, the nature of the data can be diverse and can come described in numerical, categorical, imprecisely, etc. In the case of numerical elements, a first approximation could be to categorise them so that, for example, the height of a person may be given by a range to which it belongs, as for instance [1.70, 1.90]. However, depending on how these intervals are defined, the obtained results may vary a lot. To avoid this, the use of linguistic labels such as "high" represented by a fuzzy set is a good option to represent the height of a person having at the same time a meaningful semantic to the user. Beside this, we may also have a dataset with imprecise knowledge where ordinary crisp methods cannot be directly applied.

To deal with this kind of data we introduce the concept of fuzzy transaction and fuzzy association rule defined in [4, 6].

**Definition 1** *Let $I$ be a set of items. A fuzzy transaction, $t$, is a non-empty fuzzy subset of $I$ in which the membership degree of an item $i \in I$ in $t$ is represented by a number in the range [0, 1] and denoted by $t(i)$.*

By this definition a crisp transaction is a special case of fuzzy transaction. We denote by $\tilde{D}$ a database consisting in a set of fuzzy transactions. For an itemset, $A \subseteq I$, the degree of membership in a fuzzy transaction $t$ is calculated as the minimum of the membership degree of all its items:

$$t(A) = \min_{i \in A} t(i). \tag{1}$$

Then, a fuzzy association rule $A \to C$ is satisfied in $\tilde{D}$ if and only if $t(A) \leq t(C)\ \forall t \in \tilde{D}$, that is, the degree of satisfiability of $C$ in $\tilde{D}$ is greater than or equal to the degree of satisfiability of $A$ for all fuzzy transactions $t$ in in $\tilde{D}$ .

Using this model the support and confidence measures are defined using a semantic approach based on the evaluation of quantified sentences as proposed in [4],[6]. Using the $GD$-method [6] and the quantifier $Q_M(x) = x$ the support of a fuzzy rule $A \to B$ results:

$$FSupp(A \to B) = \sum_{\alpha_i \in \Lambda(A \cap B)} (\alpha_i - \alpha_{i-1}) \frac{|(A \cap B)_{\alpha_i}|}{|\tilde{D}|} \tag{2}$$

where $\Lambda(A \cap B) = \{\alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of $\alpha$-cuts. In the previous formula, by abuse of notation, we consider the associated fuzzy set to a set of items, i.e. $X_{\alpha_i}$ represents the $\alpha$-cut of the fuzzy set derived from the itemset $X$, which is the fuzzy set with membership degree $\mu_X(t) = t(X) = \min_{i \in X} t(i)$ where $X \subset I$. Note that the elements of the fuzzy set derived from $X$ are the transactions.

Analogously the confidence is computed as follows:

$$FConf(A \to B) = \sum_{\alpha_i \in \Lambda(A \cap B)} (\alpha_i - \alpha_{i-1}) \frac{|(A \cap B)_{\alpha_i}|}{|A_{\alpha_i}|} \tag{3}$$

Employing support and confidence measures and setting appropriated thresholds for them fuzzy association rules can be discovered. In [7] it is proposed a parallelization of the computation of $FSupp$ and $FConf$ using a set of predefined $\alpha$-cuts. Note that considering a sufficiently dense set of $\alpha$-cuts in the unit interval, the obtained measure will be a good approximation of the real measure that should consider every $\alpha \in [0, 1]$ appearing in the dataset. This is the main idea of our proposal for mining fuzzy association rules using MapReduce. Firstly, the algorithm developed using MapReduce is applied repeatedly for every $\alpha$-cut. The final step consists in applying a MapReduce phase which aggregates the results using the previous formulas for support and confidence.

## 4 BDFARE Algorithm

Traditional algorithms have some problems when dealing with large amounts of data as they make multiple scans of the whole database. This means that the execution time increases with the number of transactions. In our research we have used Spark to improve Apriori algorithm for mining fuzzy association rules as follows.

The data is stored in a Big Data architecture, for which we used Hadoop (which allows for replication through its HDFS - Hadoop Distributed File System) and enables distributed processing using Spark.

This new algorithm, called BDFARE (Big Data Fuzzy Association Rules Extraction) algorithm, is based on two phases. The first one involves loading the dataset and calculating how often each item appears in the set of transactions using the Map and Reduce functions. In Figure 1 we can see an example of first phase with the value of $\alpha = 0.5$. In this example, the $Map()$ function, in particular it is used the $FlatMap()$ function, returns only the items with support higher than 0.5 (column in the middle only contains items with membership values $\geq 0.5$). After that, a re-counting of the obtained items is done using the $Reduce()$ function.

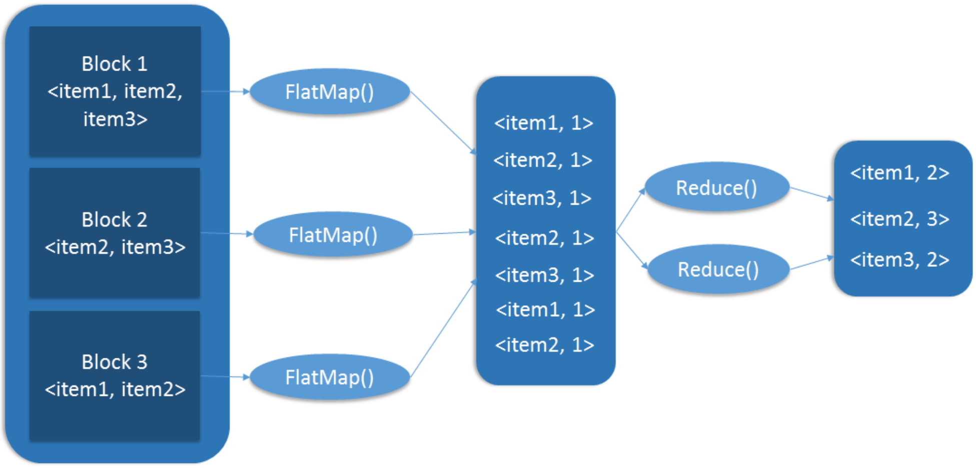


**Fig. 1.** First Phase of BDFARE

This is followed by the second phase, in which we extract the size-$k$ itemsets (see Figure 2). For this task we used a function that returns the candidate $k$-itemsets from the frequent items. As a result, a list of pairs of the type $< itemset, degree\_of\_membership >$ is saved in a global variable representing the candidate list. This enables a faster access to the list. After this, the $map()$ function returns, as in the previous phase, the counts of the candidates, having a pair $< itemset, 1 >$ when the membership value of the itemset is higher than the value of the $\alpha$ considered in that iteration. In this way, the algorithm calculates in each step all the cardinalities necessary for the computation of the final support and confidence measures. These two phases will be repeated until no new $k$-itemsets can be found.

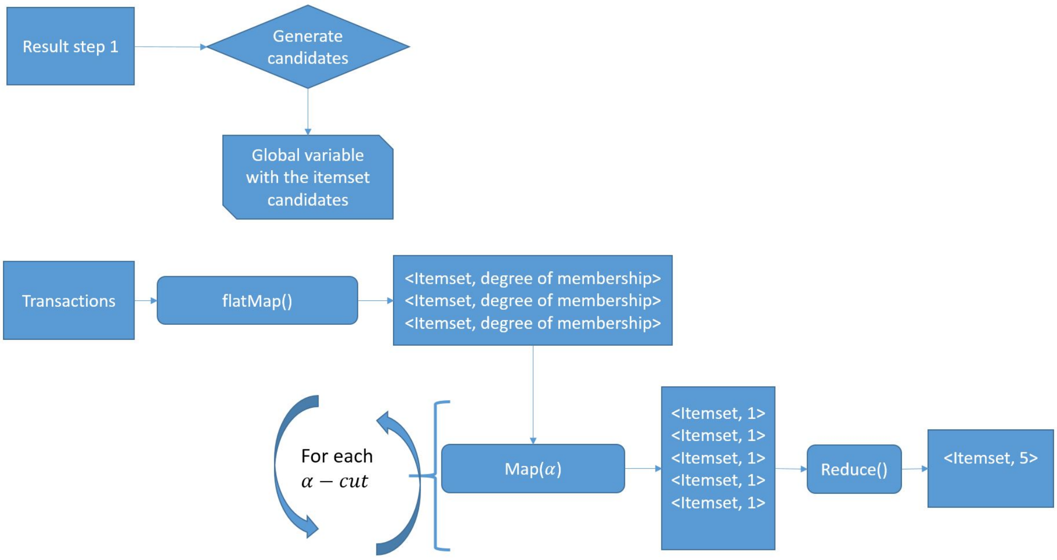


**Fig. 2.** Second Phase of BDFARE

As mentioned in Section 2, we have employed a hash table as the central data structure to accelerate the searching of itemsets. If we use instead of the hash table a linear search at each node it will result in an increase of time. In [24] a comparison between the different data structures was made concluding that the hash table outperformed among the three data structures (hash tree, trie and hash table ) for both real-life and synthetic datasets.

## 5 Experiments and results

In order to check the performance of our proposal we have carried out several experiments to compare running time using the non-distributed version of Apriori for discovering fuzzy association rules with the distributed proposal using BDFARE algorithm. Our aim is to study the behaviour of the algorithm with and without Big Data techniques. To this end, we applied the algorithms in several fuzzy transactional datasets and we study their running time according to different parameters: the number of items and the number of transactions of the datasets.
Three different datasets have been considered from the UCI machine learning repository [1] where some attributes have been conveniently fuzzyfied as described in [23]. The `German` dataset consists of transactions about credits offered by a german bank. Three variables were fuzzyfied: amount of the credit, its duration and the age of the person who owns the credit. The `Autompg` dataset consists of several attributes about cars. In this case, the continuous attributes were fuzzified using the following linguistic labels: low, medium and high. The `Bank` dataset contains data about marketing campaigns of a Portuguese banking institution. In this dataset we have fuzzified their continuous attributes by defining a suitable fuzzy partition according to the semantics of the attribute (description of the fuzzy sets employed can be found in [23]).

The final datasets used in the experiments have been replicated in order to obtain larger datasets to prove the performance of the algorithm in extreme situations. Their original size can be found in Table 1. Actually, there is not any large fuzzy dataset available in open data repositories, but we plan to apply the algorithm to data collected during a time period from sensored buildings. These sensors give numerical values from a continuous scale that are very close among them (e.g. 25 degrees and 25,2 degrees). In this case, the building operators are more interested in obtaining patterns relating temperatures such as “warm”, “cold”, “very cold”, etc. that can be represented by convenient fuzzy sets instead of using intervals that may divide very close data in two different intervals.

| Fuzzy Database | Transactions | Items |
|---|---|---|
| `German` | 1000 | 79 |
| `Autompg` | 398 | 39 |
| `Bank` | 45211 | 112 |

**Table 1.** Datasets

[1] http://archive.ics.uci.edu/ml/

### 5.1 Results

The experimental evaluation have been made in an computer architecture consisting of a cluster comprised of three processing units with Intel Xeon processors with 4 cores at 2.2 Ghz, while the traditional algorithm was executed on one of these clusters. We have performed several experiments with different thresholds values, but we show here the results for minimum support equal to 0.2 and minimum confidence equal to 0.8. The important thing here is to set the same thresholds for distributed and non-distributed approaches since we are more interested here in observing the performance (time and memory) of both approaches depending on the number of transactions and the number of items.

We began by studying the behaviour of the algorithms (distributed and non-distributed) when the number of transactions increases. To this end, they were run as subsets of the whole datasets in order to observe the behaviour with regard to the number of transactions. Figure 3 shows the behaviour of the algorithm when the quantity of transactions increases in each of the datasets (the number of transactions has been replicated till obtaining four millions). It can be observed that the traditional algorithm performed worst than the BDFARE algorithm, as expected.

To be exhaustive, with the datasets consisted of 0,5 million and 1 million transactions, we achieved time savings in average of 12% and 18% respectively compared to the non-distributed version. We can also see in this graph that the traditional algorithm did not complete its execution for the dataset containing 4 million transactions due to an error resulting from the lack of memory. By contrast, BDFARE completed the execution thanks to the use of Big Data techniques, which allow us to use the memory of the three nodes, thereby obtaining a higher capacity.

Figure 4 shows the running time when the number of items increases in the three datasets. In this graph we observe that the execution time using BDFARE does not offer substantial improvements in some cases compared to the traditional algorithm. This is because the Apriori algorithm explores all possible item combinations and in each of these explorations it consults the dataset. Additionally, with a small number of items BDFARE does not achieve always more efficient executions due to the time consumed in the planning of the jobs, necessary when distributing data. However when the number of transactions increases, the performance of the BDFARE algorithm tends to improve the non-distributed approach.

## 6 Conclusions and future research

As we have seen, this paper has focused on the study of one of the most commonly used techniques in data mining, association rules, which allows to extract co-occurrence patterns from datasets. The algorithms proposed traditionally for mining association rules fail when analysing massive datasets because the process results in very high computational costs and its efficiency decreases when

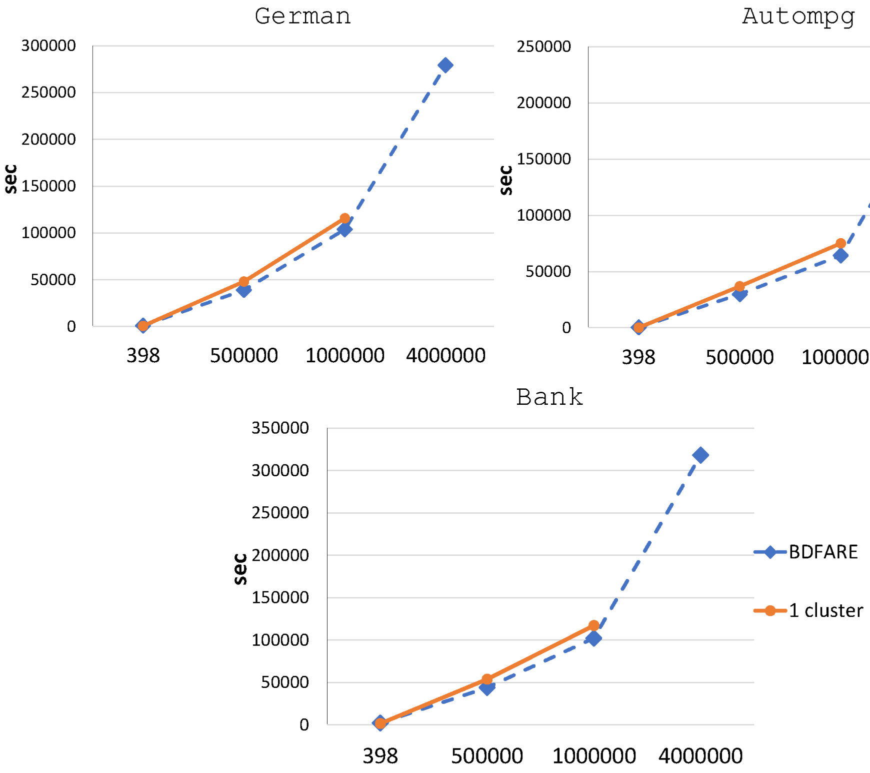


**Fig. 3.** Performance of BDFARE vs non-distributed algorithm when the quantity of transactions increases

the dataset grows.

To this end we have presented an extension to Apriori algorithm for mining fuzzy association rules using Spark, a Big Data framework which enables MapReduce implementations. The algorithm has been compared with non-distributed version of the algorithm showing improvements not only in terms of execution time and but also in terms of memory, improving the processing capacity, since some of the experiments were not able to process with the non-distributed version. An additional advantage is that our algorithm and its performance can be easily improved even further just by expanding our processing system with more clusters (computation nodes). This allows to scale our approach in external data centers or in cloud systems such as AWS (Amazon Web Services), another great advantage of Big Data technology.

As regards future research, we want to implement more efficient approaches [16] that have been proved that performs quite well in the non-distributed case such as the Apriori-TID [2], FP-Growth [20] or ECLAT [27] algorithms. Additionally, we plan to apply the presented approach and the new implementations to sensor data collected from several buildings in order to study the efficiency

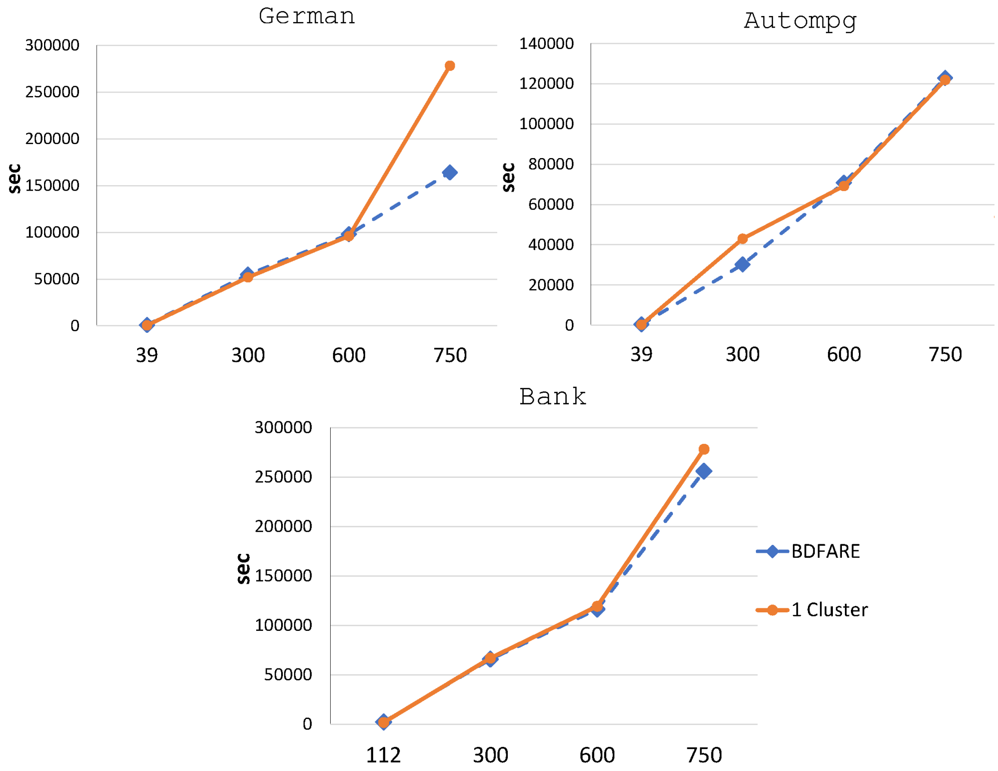


**Fig. 4.** Performance of BDFARE vs non-distributed algorithm when the quantity of items increases

patterns relating indoor and outdoor temperatures, HVAC (Heating, ventilation, air-conditioning) set points and energy consumptions. In this case, fuzzy sets are suitable to represent understandable value ranges for the users, building operators, etc.

**Acknowledgements** The research reported in this paper was partially supported by the Andalusian Government (Junta de Andalucía) under projects P11-TIC-7460 and the Spanish Ministry for Economy and Competitiveness by the project grant TIN2015-64776-C3-1-R.

## References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining associations between sets of items in large databases. In *ACM-SIGMOD International Conference on Data*, pages 207–216, 1993.

2. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the Twentieth International Conference on Very Large Databases*, pages 487–499, Santiago, Chile, 1994.
3. D.C. Anastasiu, J. Iverson, S. Smith, and G. Karypis. *Frequent Pattern Mining*, chapter Big Data Frequent Pattern Mining, pages 225–259. Springer International Publishing Switzerland, 2014.
4. F. Berzal, M. Delgado, D. Sánchez, and M.A. Vila. Measuring accuracy and interest of association rules: A new framework. *Intelligent Data Analysis*, 6(3):221–235, 2002.
5. S. del Río, V. López, J. M. Benítez, and F. Herrera. On the use of mapreduce for imbalanced big data using random forest. *Information Sciences*, 285:112 – 137, 2014. Processing and Mining Complex Data Streams.
6. M. Delgado, N. Marín, D. Sánchez, and M.A. Vila. Fuzzy association rules: General model and applications. *IEEE Transactions on Fuzzy Systems*, 11(2):214–225, 2003.
7. M. Delgado, M.D. Ruiz, D. Sánchez, and J.M. Serrano. A formal model for mining fuzzy rules using the RL representation theory. *Information Sciences*, 181(23):5194–5213, 2011.
8. X. Meng et al. Mllib: Machine learning in apache spark. arXiv preprint: abs/1505.06807, 2015.
9. Z. Farzanyar and N. Cercone. Accelerating frequent itemset mining on the cloud: A mapreduce-based approach. In *IEEE 13th International Conference on Data Mining Workshops*, pages 592–598, 2013.
10. Z. Farzanyar and N. Cercone. Efficient mining of frequent itemsets in social network data based on mapreduce framework. In *Proceedings of the 2013 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, pages 1183–1188, 2013.
11. A. Fernández, C.J. Carmona, M.J. del Jesus, and F. Herrera. A view on fuzzy systems for big data: Progress and opportunities. *International Journal of Computational Intelligence Systems*, 9:69–80, 2016.
12. C. Fernandez-Basso, M.D. Ruiz, and M.J. Martin-Bautista. Extraction of association rules using big data technologies. *Int. J. of Design & Nature and Ecodynamics*, 11(3):178–185, 2016.
13. M. Gabroveanu, M. Cosulschi, and N. Constantinescu. A new approach to mining fuzzy association rules from distributed databases. *Annals of the University of Bucharest*, LIV:3–16, 2005.
14. M. Gabroveanu, M. Cosulschi, and F. Slabu. Mining fuzzy association rules using mapreduce technique. In *International Symposium on INnovations in Intelligent SysTems and Applications*, INISTA, pages 1–8, 2016.
15. M. Gabroveanu, I. Iancu, M. Cosulschi, and N. Constantinescu. Towards using grid services for mining fuzzy association rules. In *Proceedings of the 1st East European Workshop on Rule-Based Applications*, RuleApps, pages 507–513, 2007.
16. Jochen Hipp, Ulrich Güntzer, and Gholamreza Nakhaeizadeh. Algorithms for association rule mining - a general survey and comparison. *ACM sigkdd explorations newsletter*, 2(1):58–64, 2000.
17. E. Hüllermeier and Y. Yi. In defense of fuzzy association analysis. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 37(4):1039–1043, 2007.
18. Haoyuan Li, Yi Wang, Dong Zhang, Ming Zhang, and Edward Y Chang. PFP: parallel fp-growth for query recommendation. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 107–114. ACM, 2008.

19. N. Li, L. Zeng, Q. He, and Z. Shi. Parallel implementation of apriori algorithm based on mapreduce. In *Proceedings of the 2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, SNPD '12, pages 236–241, Washington, DC, USA, 2012. IEEE Computer Society.
20. J. Pei, Y. Yin, R. Mao, and J. Han. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data mining and knowledge discovery*, 8(1):53–87, 2004.
21. Hongjian Qiu, Rong Gu, Chunfeng Yuan, and Yihua Huang. Yafim: A parallel frequent itemset mining algorithm with spark. In *Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International*, pages 1664–1671. IEEE, 2014.
22. S. Rathee, M. Kaul, and A. Kashyap. R-apriori: An efficient apriori based algorithm on spark. In *Proceedings of the PIKM'15*, Melbourne, VIC, Australia, 2015. ACM.
23. M. D. Ruiz, D. Sánchez, M. Delgado, and M. J. Martin-Bautista. Discovering fuzzy exception and anomalous rules. *IEEE Transactions on Fuzzy Systems*, 24(4):930–944, 2016.
24. S. Singh, R. Garg, and P.K. Mishra. Performance analysis of apriori algorithm with different data structures on hadoop cluster. *International Journal of Computer Applications*, 128(9):45–51, 2015.
25. Tom White. *Hadoop: The Definitive Guide. Fourth edition.* O'Reilly, 2015.
26. L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
27. Mohammed Javeed Zaki. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(3):372–390, 2000.

## 2.2 Spark solutions for discovering fuzzy association rules in Big Data(Submitted to Applied Soft Computing)

- Carlos Fernandez-Basso, M. Dolores Ruiz, Maria J. Martin-Bautista.
  - Status: **Submitted to Applied Soft Computing**.
  - Impact Factor (JCR 2018): **4.873**
  - Subject Category: **Computer Science, Artificial intelligence**
  - Rank: **20/134**
  - Quartile: **Q1**

# Spark solutions for discovering fuzzy association rules in Big Data

Carlos Fernandez-Basso[a], M. Dolores Ruiz[b], Maria J. Martin-Bautista[a]

[a]*Department of Computer Science and A.I. and CITIC-UGR, University of Granada*
[b]*Department of Statistics and Operations Research, University of Granada*

**Abstract**

The high computational impact when mining fuzzy association rules grows significantly when managing very large data sets, triggering in many cases a memory overflow error and leading to the experiment failure without its conclusion. It is in these cases when the application of Big Data techniques can help to achieve the experiment completion.

Therefore, in this paper several Spark algorithms are proposed to handle with massive fuzzy data and discover interesting association rules. For that, we based on a decomposition of interestingness measures in terms of $\alpha$-cuts, and we experimentally demonstrate that it is sufficient to consider only 10 equidistributed $\alpha$-cuts in order to mine all significant fuzzy association rules.

Additionally, all the proposals are compared and analysed in terms of efficiency and speed up, in several datasets, including a real dataset comprised of sensor measurements from an office building.

*Keywords:* Big Data, Fuzzy frequent itemset, Fuzzy Association Rules, Spark

Nowadays, the increasing data generation in the majority of companies, social media, etc. has given rise to the Big Data phenomenon. Moreover, every daty more buildings sum to the “smart sensored” fashion by incorporating sensor devices to collect data in order to use it for improving their energy usage and therefore save not only energy but also money and natural resources. The

*Email address:* `cjferba@decsai.ugr.es` (Carlos Fernandez-Basso[a])

constantly necessity of information discovery from these huge amounts of data is the key in this competitive world. In this regard, Big Data philosophy, based on the MapReduce framework, helps in that task. In particular, many Data Mining techniques had been developed using the Big Data techniques [1, 2, 3, 4]. Association rules can be helpful in this ambit to uncover hidden relationships in gathered data without supervising. Association rules are often represented by implications of the type $A \rightarrow B$, representing the joint co-occurrence of $A$ and $B$ in a high percentage of transactions. However, numerical data coming e.g. from sensors may contain very granulated values that difficult their analysis. At this respect, some works have employed discretization of these numerical values. But this may biassed the obtained results since final results can vary a lot depending on how the attribute values are divided. This problematic has been pointed out by several authors like in [5]. Fuzzy Sets theory [6] has been proven to be a good option to discretized numerical values in a soften way and representing in a understandable way to users. As a result of this fuzzy discretization process, fuzzy databases are created where fuzzy association rules can be extracted [7]. Additionally, there are ambits where data can be affected by some kind of uncertainty or imprecision, and they are gathered following a fuzzy description of attributes [8].

In this paper, we study which can be the best procedure to analyse Big Data by means of fuzzy association rules. In the literature, some proposals can be found to perform such analysis, but they run into problems when the volume of data increases, becoming less efficient and leading, in many cases, to memory overflow errors. To this end, we have proposed different Spark proposals for mining fuzzy association rules, enabling the distributed processing of big data amounts. Spark implementations [9] enables faster memory operations than other framework like Hadoop [10] because in-memory computations are allowed. In [11] there is a very complete comparison between Hadoop and Spark frameworks in several Machine Learning algorithms, obtaining that Spark outperforms Hadoop in the majority of cases.

Fuzzy association rule in the Big Data environment has not attracted yet many attention, and to the best of our knowledge, there does not exist any implementation. Therefore, the main contribution of this work is then the proposal, implementation and comparison of different algorithms for fuzzy association rule mining in Spark, improving the existing implementations of crisp association rule algorithms, by enabling the uncovering of fuzzy association rules with no restrictions in either the number of items present in the antecedent or the consequent. The results of our experiments show that Big Data approach improves the efficiency of algorithms, with respect to traditional non-distributed techniques. For that we have compared their performance in terms of execution time and also in memory consumption, attending to the number of transactions and the number of items to be analysed. However, the growth in the number of items, due to the exponential combination among them, the different proposals may not achieve a significant improvement in time, but thanks to the distribution capability of Big Data, substantial improvements are achieved in terms of memory problems, enabling to finish the analysis, in contrast to the memory overflow problems that sometimes appear in traditional approaches.

Additionally, our proposals are based on a decomposition of interestingness measures in terms of $\alpha$-cuts which facilitates their implementation to other interestingness measures different to that of support and confidence using the formal model developed in [12]. In order to facilitate the decision of what set of $\alpha$-cuts it is necessary to employ when mining fuzzy association rules, we experimentally demonstrate that it is sufficient to consider only 10 equidistributed $\alpha$-cuts in order to uncover all significant fuzzy association rules.

The paper organization is the following: Section 1 makes a literature revision in the ambit of Big Data technologies and association rule mining algorithms. Section 2 introduces the definitions and concepts related to Big Data and fuzzy association rule mining necessary for the comprehension of the paper. Section 3 presents the different algorithms proposed and developed for fuzzy association rules mining using Spark. Section 4 describes the experiments performed and results obtained, before concluding the paper in Section 5.

## 1. Related work

Data mining techniques can be classified into two main types: supervised techniques such as classification methods and non-supervised techniques such as clustering or association rules. This work is focused in association rule mining, but there are also many works which restrict to frequent itemset mining, the first step in the process of discovering association rules.

### *1.1. Algorithms review for frequent itemset mining*

In this section we review the existent algorithms in the literature for frequent itemset extraction. Once frequent itemsets are discovered, association rules can be extracted by assessing the strength of the relation (e.g. by means of confidence).

*Apriori based algorithms*

Apriori algorithm was proposed in the mid-nineties [13]. It consists on finding the set of frequent itemsets $L$, in a given database $D$. This algorithm searches the items in the transactions structure, which is consulted in each iteration to check if larger itemsets are satisfied. In this first work [13], the authors identified a fundamental property: the downward closure property. This property assures that any subset of a frequent itemset must be also frequent. This gave rise to divide the process in two main steps that are repeated to find frequent itemsets of length $k$: first one is known as the candidate generation step, in which the support of the corresponding $k$-itemsets is calculated by scanning the transactional database, and second one is known as the large itemset generation, in which frequent $k + 1$-itemsets are generated by pruning the candidate itemsets that do not exceed the minimum support threshold. These steps are then repeated until no more frequent itemsets can be found. Afterwards, an interestingness measure like the confidence is employed to discover the association rules, using the frequent itemsets extracted in the first step.

There are two limitations of this algorithm: one limitation comes from the complex itemset generation process which is very costly in time and memory, and the second limitation comes from the excessive number of scans necessary in the candidate generation step.

An improvement of Apriori is the algorithm called Apriori-TID[13]. This version of Apriori algorithm improves the performance of traditional Apriori algorithm with large databases. This approach is based on an itemset reduction by removing non-frequent itemsets in the database in each step, and has some changes with respect to the Apriori algorithm. The first change consists of sorting the transactions by item frequency, and removing the non-frequent items. The second change is that it uses the previous $k$ candidate itemsets which were calculated in the previous phase to generate the itemsets of $k + 1$ size.

This version improves the Apriori algorithm in small datasets, but for large datasets the performance of Apriori-TID is similar to Apriori although the use of memory is improved [14].

*ECLAT TID-list based algorithm*

Another way to improve the frequent itemset mining is by means of intermediate storage of data. In this case, a TID-list[1] is employed [15], where a binary list is created for each itemset, containing 1 in position $j$, if the item is satisfied in the transaction $j$ and 0 otherwise[2]. This structure allows the computation of item and itemset support using the boolean operators. The use of these operations improves the performance of the algorithm because boolean operations are performed very efficiently. One of the most known algorithms using this approach is the ECLAT [14][16] algorithm.

The main problem of this algorithm is that for a large number of items it would be necessary to store very large lists and the memory consumption would grow a lot. Additionally, the extension of this process in the distributed environment is not straightforward because each TID-List has a lot of dependencies with other TID-lists. However, the performance of ECLAT is faster in comparison with

[1] TID stands for Transaction IDentifier.

[2] Another variant of TID-list consists on a list for each item containing the ids of the transactions where the item is satisfied.

Apriori in the classical framework [17].

*Frequent Pattern FP-Growth algorithm*

Another kind of algorithm for frequent itemset extraction is that using the FP-tree structure [18]. This algorithm called FP-Growth [19] uses the divide-and-conquer technique. The FP-Growth decomposes the search space based on length-1 suffixes and reduces the number of searches in the database by using the representation of the data in a FP-tree structure.
This algorithm was originally designed for query recommendation where only top-k frequent itemsets are extracted. This issue is a drawback for association rule mining, because frequent itemsets of every length are needed to extract the association rules, unless we are interested in only the association rules with higher support. This means that FP-Growth is not exhaustive.

*Algorithm comparison*

In the literature different comparative studies of frequent itemset algorithms can be found [20]. Amongst them we can emphasize works comparing the Apriori algorithm most widely used and known, to the rest of approaches. In [17] the Apriori algorithm is compared with ECLAT and it was appreciated how the use of the TID-lists improves the performance of the Apriori, although the use of memory is greater. Regarding the FP-Growth algorithm it scans the database of transactions only once, having thus a faster algorithm than Apriori [20]. However, one of the problems of FP-Growth is that, for very large datasets, the FP-tree may not fit in memory. Another feature to take into account, is that, FP-Growth is not exhaustive, that is, it does not obtain all the possible association rules since it does not generate all the possible frequent itemsets.
Moreover, in [21] there is a study comparing the most employed approaches, namely Apriori, ECLAT and FP-Growth. The experimental evaluation concludes that FP-Growth is more scalable and outperforms the others. In [22], a comparative study in the Big Data paradigm in a crisp framework is presented. As a conclusion, it shows that the distributed adaptation of FP-Growth is not

always convenient to extract association rules, since it only provides the most frequent itemsets and their support. In our case, the algorithms of our proposal are exhaustive, finding all possible frequent itemsets. Hence, we propose the implementation and comparison of the Apriori, AprioriTID and ECLAT based algorithms in Spark to extract both frequent itemsets and fuzzy association rules in an exhaustive way.

### *1.2. Big Data algorithms for frequent itemset mining*

The most famous framework employed in Big Data was designed by Google in 2003, called *MapReduce*. The MapReduce framework foundation lies on two different functions, as its name indicates, to distribute the computation. On one hand, the *Map()* function is in charge of transforming data into pairs of the type $(key, value)$ attending to some criteria that should be specified. And the *Reduce()* function is employed to aggregate those key-value pairs sharing the same key to finally obtain a piece of processed data according to the specified criteria.

When implementing MapReduce algorithms, two different frameworks arise as the most employed: Hadoop and Spark. These platforms have been improved in the last few years by incorporating diverse functions to fully take advantage of the capacity processing of a cluster, enabling to obtain thus more scalable algorithms and improving substantially traditional ways of cluster programming. In particular, for the case or association rules, within the Spark library, the PFP (Parallel FP-Growth) is included. This is a distributed adaptation of known FP-Growth algorithm that can be employed to extract higher level itemsets exceeding the minimum support threshold [23].

Other proposals for mining frequent itemsets using MapReduce techniques can be found in the literature for the non-fuzzy case. We can highlight some approaches which present Apriori-base algorithms using Hadoop: [24, 25, 26]. Note that these implementations are made in Hadoop, and according to the analysis made in [11], Spark accelerates executions since it enables in memory computations. Moreover, algorithms proposed in [24, 25, 26] do not employ data structures like tree, hash tree or hash table that can help to decrease execution times (see the study made in [27]).

In [28] and [29] the authors proposed the R-Apriori and YAFIM algorithms respectively, which are Apriori-based algorithms using Spark. These proposals can be compared to the non-fuzzy phase of our approach (made for each $\alpha$-cut), but their proposal bases on a vectorial processing to obtain the itemsets of length $k$ in the distributed process using a hash tree, while our proposal applies MapReduce for every $k$-itemset using a hash table. In addition, posterior analysis made in [27] which compares MapReduce implementations for different data structures concluded that using the hash table accelerates the algorithm performance versus using hash trees and tries (prefix trees).

Beside this, parallel adaptations of most used algorithms have been also developed for frequent itemset mining [30] such as: ParEclat (Parallel Eclat) [31], Par-FP (Parallel FP-Growth with Sampling) [32], HPA (Hash Partitioned Apriori)[33]. All these algorithms use different data structures to improve the performance and take advantage of the potential of multiprocessors. But the problem is that the multiprocessor it is not enough for all cases. When data exponentially increase, algorithms need more processing capability. For this reason distributed algorithms arise as a new option for frequent itemset extraction.

### *1.3. Distributed algorithms for association rule mining*

As described previously, most of the reviewed algorithms focus only on frequent itemset data extraction. Besides, in the literature as far as we know, no significant improvements have been described in the association rule creation phase, since the most time/memory consuming part is that of obtaining frequent itemsets. We can highlight the work of [34] where PEAR (Parallel Efficient Association Rules) algorithm is developed to improve the association rule mining step, but the global complexity is not decreased because the cost of extracting the frequent itemsets is much higher than in other algorithms. In [23] it is developed a distributed algorithm which extracts association rules but enabling only one item in the consequent part of the rule.

Conversely, in our approach, described in Section 3, it is presented an improvement of this part that enables the uncovering of association rules with no restrictions in either the number of items in the antecedent or the consequent.

### *1.4. Distributed algorithms for fuzzy association rule mining*

The best to our knowledge, the only one work using the MapReduce framework for mining fuzzy association rules can be found in [35]. This proposal is based on an extension of the Count Distribution algorithm [36, 37] to the fuzzy case. This algorithm uses similar procedure than R-Apriori [28] algorithm where in the second phase, in charge of computing the itemset support, Hadoop is employed instead of Spark. This approach differs from our proposal since we employ the resilient distributed dataset structure that enables across cluster computation.

## 2. Preliminaries

### *2.1. Fuzzy association rules*

Agrawal et al. [38] formally defined association rules for the first time, although in Observational Calculi [39, 40] it was also investigated the analysis of associations.

In general, the Association Rule discovery problem consists in uncovering implications of the form $A \rightarrow B$ where $A, B$ are subsets of items from $I = \{i_1, i_2, \ldots, i_m\}$ fulfilling that $A \cap B = \emptyset$ in a database formed by a set of $n$ transactions $D = \{t_1, t_2, \ldots, t_n\}$ each of them containing subsets of items from $I$. $A$ is usually referred as the antecedent and $B$ as the consequent of the rule.

The problem of association rules discovery has two differentiated sub-tasks consisting on

- finding all the itemsets exceeding the imposed threshold for the support, where support is defined as the percentage of transactions containing or satisfying an itemset. These are known as frequent itemsets.

- Once frequent itemsets are obtained, association rules are those exceeding the minimum confidence or another established assessment measurement (e.g. Lift).

However, as it was mentioned in the introduction, the data can be diverse and can come described numerically, categorically, imprecisely, etc. For continuous numerical attributes, it is often applied a categorization process, for example, the price of an object may be given by a range to which it belongs, as for instance [100, 200]. However, depending on the definition of the intervals, the obtained associations can vary a lot. To avoid this, fuzzy linguistic labels appear as a solution to overcome this problem. In the previous example, a label like "expensive", that can be represented by a fuzzy set, is a good option to represent the price of an object having at the same time a meaningful and understandable semantic to the user [5]. Moreover, there can be occassions where ordinary crisp methods for describing data cannot be directly applied (see for instance [41]).

In all these cases, the data has to be represented by a gradual value, leading to the concept of fuzzy transaction and fuzzy association rule. Definitions of these concepts are taken from [42, 7].

**Definition 1.** *A fuzzy transaction, $t$, is a non-empty fuzzy subset of $I$, where $I$ is a set of items. That is, the membership degree of an item $i \in I$ in $t$ is represented by a number in the range [0, 1] and denoted by $t(i)$.*

Note that this definition generalizes the idea of crisp transaction to the special case of fuzzy transaction. From now on, $\tilde{D}$ will denote a fuzzy transactional database (see for instance Table 1).

**Definition 2.** *Let $A$ denote an itemset, i.e. a subset of items in $I$. The degree of membership of $A$ in a fuzzy transaction $t \in \tilde{D}$ is defined as follows:*

$$t(A) = \min_{i \in A} t(i). \tag{1}$$

*This means that $t(A)$ is the minimum of the membership degree of all its items.*

**Definition 3.** *Let $A, B$ be itemsets ($\subset I$) in a fuzzy database $\tilde{D}$. Then, a fuzzy association rule $A \to B$ is satisfied in $\tilde{D}$ if and only if $t(A) \leq t(B) \, \forall t \in \tilde{D}$, that is, the degree of satisfiability of $B$ in $\tilde{D}$ is greater than or equal to the degree of satisfiability of $A$ for all fuzzy transactions $t$ in $\tilde{D}$ .*

Assessment measures for fuzzy association rules have been studied and analysed from different perspectives (a good review can be found in [43]). The cardinality based generalization proposed in [7] and also generalized in works like [44, 45], is a good option due to their good properties (see [7] and [**?** ]). However, other measures arise using of the combination of particular inclusion and cardinality operators. The study made in [46] focuses in studying the suitable operators yielding a fuzzy Ruspini's partition in close relation with negated items in rules. For a deeper discussion on the possible frameworks to assess fuzzy association rules we recommend the review made in [43].

The support and confidence measures employed in this work are based on a semantic approach for the evaluation of quantified sentences [42] using the *GD*-method [7] and the quantifier $Q_M(x) = x$, which represents the quantifier *"the majority"*. The following definitions can be found in [42, 7].

**Definition 4.** *The support of a fuzzy itemset $A$ is defined as:*

$$FSupp(A) = \sum_{\alpha_i \in \Lambda} (\alpha_i - \alpha_{i+1}) \frac{|\{t \in \tilde{D} : t(A) \geq \alpha_i\}|}{|\tilde{D}|} \tag{2}$$

*where $\Lambda = \{\alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of $\alpha$-cuts.*

**Definition 5.** *The support of a fuzzy association rule $A \to B$ is defined as:*

$$FSupp(A \to B) = \sum_{\alpha_i \in \Lambda} (\alpha_i - \alpha_{i+1}) \frac{|\{t \in \tilde{D} : t(A) \geq \alpha_i \text{ and } t(B) \geq \alpha_i\}|}{|\tilde{D}|} \tag{3}$$

*where $\Lambda = \{\alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of $\alpha$-cuts.*

**Definition 6.** *The confidence of a fuzzy association rule $A \to B$ is defined as:*

$$FConf(A \to B) = \sum_{\alpha_i \in \Lambda} (\alpha_i - \alpha_{i+1}) \frac{|\{t \in \tilde{D} : t(A) \geq \alpha_i \text{ and } t(B) \geq \alpha_i\}|}{|\{t \in \tilde{D} : t(A) \geq \alpha_i\}|} \tag{4}$$

*where $\Lambda = \{\alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of $\alpha$-cuts.*

Fuzzy association rules can be discovered by fixing a set of predefined $\alpha$-cuts [44] to compute the support and confidence measures. Note that with a sufficiently dense set of $\alpha$-cuts in the unit interval the computed measures will be closer to the real measure obtained by considering every $\alpha \in [0, 1]$ appearing in the dataset. This idea is behind of our proposal using MapReduce for mining fuzzy association rules. And we will see in the experiments that it is enough to consider 10 equidistributed $\alpha$-cuts in order to obtain the whole set of fuzzy association rules.

## 3. Fuzzy Association Rule Mining Algorithms using Spark

In this section, we present three new algorithms for frequent itemset extraction and the common part of these algorithms for fuzzy association rules (from now on FAR) mining, all of them following the Big Data paradigm for distributed-mode algorithms. These proposals are inspired by the operation of the traditional sequential Apriori, Apriori-TID and ECLAT algorithms and are all implemented using the Spark Framework, which has several facilities for developing Big Data algorithms based on MapReduce and improvements like the use of main memory or advanced DAG. In this regard, the implemented data structure in Apache Spark, called Resilient Distributed Dataset (RDD), abstracts the concept of data partition and will be employed through all our proposals, meaning that data are distributed across the clusters [47].

Before presenting the algorithms it is necessary to describe the following primitive Spark functions:

- *Map*: Applies a transformation function to each RDD and returns a transformed RDD. For instance, a Map function applied to $< key, value >$ pairs will give transformed pairs:

$$Map(< item, value_i >) \rightarrow < item, transformed\_value_i > \qquad (5)$$

- *FlatMap*: Similar to *Map*, but each input item can be mapped to 0 (Equation 6 A), or to a pair (Equation 6 B) or different output items by means of auxiliary functions (Equation 6 C).

$$FlatMap(< item, value_i >) \rightarrow \emptyset \tag{6}$$

$$FlatMap(< item, value_i >) \rightarrow < itemset, value_j > \tag{7}$$

$$FlatMap(< item, value_i >) \rightarrow set(< itemset, value_j >) \tag{8}$$

- *Reduce*: Aggregates the elements of the dataset using an aggregation function. For example, the frequency of appearance of an item is computed by applying a *Reduce* function in the following way:

$$Reduce(< item, list(value) >) \rightarrow < item, value_{aggregated} > \tag{9}$$

- *Filter*: This function allows to filter the distributed data according to a condition.

Additionally, the algorithms use broadcast variables to enable access to global variables in every node of the cluster, i.e. broadcast variables are available in every partition performed by the Map functions.

In the following, we explain the different approaches proposed to extract frequent itemsets and fuzzy association rules using Big Data technologies.
Traditional algorithms problems when dealing with large amounts of data are mainly due to the multiple scans made of the whole database. This often raise in an increasing of the execution time along with the number of transactions. In our proposals, Spark is employed to improve the Apriori and ECLAT algorithms for mining fuzzy association rules. In both cases the data is stored using the HDFS (Hadoop Distributed File System), which permits replication and enables distributed processing.
In the following sections we present the different algorithms designed for extracting fuzzy association rules using Spark.

### *3.1. BDFARE-Apriori*

This section is devoted to present the new algorithm, BDFARE-Apriori (Big Data Fuzzy Association Rules Extraction), which is comprised of different phases: preprocessing, phase 1, phase 2 and fuzzy association rule extraction.

The overall process is depicted in Algorithm 1. In that, we have employed the acronym DCS (Distribute Computing using Spark) to representing each chunk of data automatically created be Spark when distributing the data among the clusters. Each chunk is noted by $S_i$. This notation is also used in the subsequent algorithms.

For a better understanding of the running algorithms we will use the example of database in Table 1.

| ID | A | B | C | D |
|---|---|---|---|---|
| 1 | 0.75 | 0.15 | 0.35 | 0.1 |
| 2 | 0 | 0.5 | 0.2 | 0 |
| 3 | 0.8 | 0.4 | 0 | 0.45 |
| 4 | 1 | 0.25 | 0.85 | 1 |
| 5 | 0.5 | 1 | 0 | 0.8 |
| 6 | 0.3 | 0 | 0.75 | 0 |

Table 1: Fuzzy database example

#### *3.1.1. Preprocessing*

The different algorithms proposed for fuzzy association rule mining algorithms pre-process the data transforming them into an array of BitSets. Other works like [48] and [49], which use this kind of representation, have obtained very good results in terms of memory usage and execution time. The BitSet representation has the advantage of accelerating logical operations such as conjunction or cardinality, which is a fundamental part of our algorithm when calculating item conjunctions and their frequencies.

Therefore, for fuzzy association rule mining, the algorithm processes the data and store them into an array of bit-lists whose size will depend on the number

**Algorithm 1** Main Spark procedure for BDFARE-Apriori algorithm

```
Input: Data: Fuzzy RDD transactions: $\{t_1, \ldots, t_n\}$
Input: AlphaCuts: List of alpha cuts: $\{\alpha_1, \ldots, \alpha_p\}$
Input: MinSupp: minimum support threshold.
Output: Frequent itemsets exceeding $MinSupp$ (FreqItemset)
                    Preprocessing
DCS in $q$ chunks of Data: $\{S_1, \ldots, S_q\}$
    $BitArray_{S_i} \leftarrow S_i$.Map $(FuzzyToArray(t_k \in S_i))$ # Map function computes independently each transaction in $S_i$
                    Phase 1: FreqItems()
DCS in $q$ chunks of BitArray: $\{BA_1, \ldots, BA_q\}$
    $\{< it_1, bit\_list_1 >, \ldots, < it_m, bit\_list_m >\} \leftarrow BA_j$.FlatMap()
    $\{< it_1, card_1 >, \ldots, < it_m, card_m >\} \leftarrow$ ReduceByKey$(|bit\_list_{it_k}|)$
    ItemSupport $\leftarrow$ Support$(it_k)$
    DicFreqItemset $\leftarrow$ Filter$(ItemSupport \geq MinSupp)$

                    Phase 2: Candidate generation
$Candidate \leftarrow DicFreqItemset$ #Candidates of $Length = 1$
$Length = 2$
$Global\_FreqItemset \leftarrow Candidate$
$broadcast(Global\_FreqItemset)$ #Creates a broadcast variable for its use across the cluster
do
    DCS in $q$ chunks of BitArray: $\{BA_1, \ldots, BA_q\}$
        $\{< itemset_1, bit\_list_1 >, \ldots, < itemset_m, bit\_list_m >\} \leftarrow$
        $BA_j$.FlatMap$(BitListComputation(Global\_FreqItemset))$
        #see Alg. 3
        $\{< itemset_1, card_1 >, \ldots, < itemset_t, card_t >\} \leftarrow$
        ReduceByKey$(|bit\_list_{itemset_k}|)$
        ItemsetSupport $\leftarrow$ Support$(itemset_k)$
        $DicFreqItemset =$ Filter$(ItemsetSupport \geq MinSupp)$
    end DCS computation
    $Length++$
    $Candidate \leftarrow CandidateGen(DicFreqItemset, Length)$
    $Global\_FreqItemset.append(Candidate)$
while $|DicFreqItemset| > 1$
return $Candidate$
```

of transactions contained in the chunk, obtaining for each transaction an array of itemsets with their corresponding bit-list. This is described in lines 6-10 of Algorithm 1. Note that all this process is made in a distributive manner, i.e. for each chunk of data, in order to accelerate the computation. Then, for each transaction and for each item a bit-list will be created containing 1 in position $j$ if the membership value of the item in that transaction is higher or equal than $\alpha_j$ and 0 otherwise. In this way, each item is represented by its bit-list depending on its value in the transaction. For that, we implement a procedure called $FuzzyToArray$, whose pseudocode is in Algorithm 2. For instance, if the itemset $X$ is satisfied with degree 0.25 in a transaction $t_i$, i.e. $\mu_X(t_i) = 0.25$, and the set of $\alpha$-cuts is $\{1, 0.75, 0.5, 0.25\}$, then the associated bit-list of $X$ in that transaction will be $[0, 0, 0, 1]$

**Algorithm 2** Pseudocode of FuzzyToArray function

```
Input: Data: a fuzzy transaction $t_k$.
Input: AlphaCuts: List of alpha cuts: $\{\alpha_1, \ldots, \alpha_p\}$
Output: An array of Bit-lists for each item
                              FuzzyToArray()
For every item $it \in t_k$
    $j = 0$
    $Array = [\ ]$
    Do
        If($\mu_{it}(t_k) \geq \alpha_j$)
            Array.append(1)
        Else
            Array.append(0)
        $j + +$
    While $j \leq p$
end for
return $\{Array\}$ # For each $t_k$ the Array contains lists of the type <item,bit_list>
```

The implementation of this type of bit-lists have been made using the *numpy* library available in python, which enables concurrent management of lists without going through every list element, enabling the distribution of computations along different clusters. Regarding the use of BitSets, it is necessary to emphasize the low memory resources needed to manage the array of bit-lists created during the preprocessing phase for big amounts of data (see also [12, 48, 49]). In addition, this structure improves the performance of the algorithm since it accelerates conjunction and cardinality computations.

### *3.1.2. Phase 1*

The first step involves the dataset load and computation of each item appearance in the set of transactions using the Map and Reduce functions. In lines 7-11 of Algorithm 1 we can see how is the process of counting the items using the MapReduce paradigm. Firstly, we use a FlatMap function (see line 8) to transform the lists $[item_1\_bit\_list], \ldots, [item_m\_bit\_list]$ to lists of pairs of the form $< item_i, bit\_list_i >$. We perform this transformation for being able to use a $ReduceByKey$ function that adds all the Bitlists of the same key (i.e. of the same item) to obtain the frequencies of each item. In Figure 1 we can see an example for the value of minimum support threshold of 0.5. In this example, the $FlatMap()$ function, returns the pairs $< item_i, bit\_list_i >$. Afterwards, the items are grouped using the $ReduceByKey$ function using the operator of *numpy* library explained previously, and their support is calculated using the formula in equation (2). The last step is in charge of selecting those items with support greater than threshold ($MinSupp = 0.5$). In the example provided in Figure 1 only item $A$ fulfils that condition.

### *3.1.3. Phase 2*

This is followed by the second phase, in which the algorithm extracts the size-$k$ frequent itemsets (lines 12-26 of Algorithm 1). For this task we use different functions, for example the $CandidateGen()$ function generates the itemsets combinations for next $k$ iteration of the algorithm. On the other hand, the function $BitListComputation()$ (see Algorithm 3) performs in a distributed way, the $AND$ combination in each transaction of the candidate itemsets stored in the global variable $Global\_FreqItemset$ through its use in a FlatMap() function (see

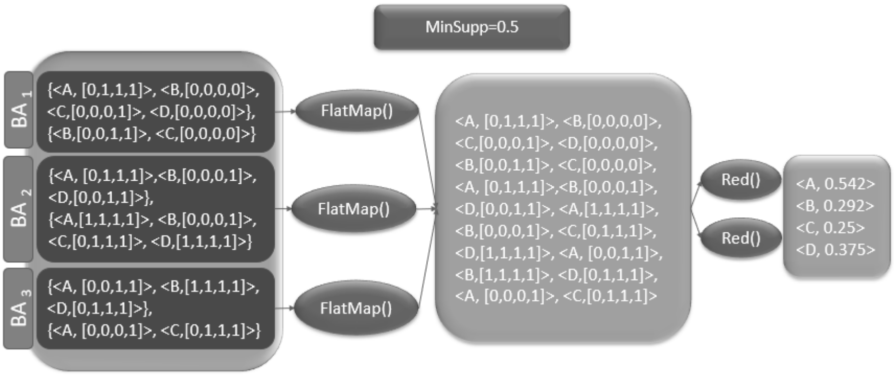

MinSupp=0.5
BA 1
{<A, [0,1,1,1]>, <B,[0,0,0,0]>, <C,[0,0,0,1]>, <D,[0,0,0,0]>}, {<B,[0,0,1,1]>, <C,[0,0,0,0]>}
BA 2
{<A, [0,1,1,1]>,<B,[0,0,0,1]>, <D,[0,0,1,1]>}, {<A,[1,1,1,1]>, <B,[0,0,0,1]>, <C,[0,1,1,1]>, <D,[1,1,1,1]>}
BA 3
{<A, [0,0,1,1]>, <B,[1,1,1,1]>, <D,[0,1,1,1]>}, {<A, [0,0,0,1]>, <C,[0,1,1,1]>}
FlatMap()
FlatMap()
FlatMap()
<A, [0,1,1,1]>, <B,[0,0,0,0]>, <C,[0,0,0,1]>, <D,[0,0,0,0]>, <B,[0,0,1,1]>, <C,[0,0,0,0]>, <A, [0,1,1,1]>,<B,[0,0,0,1]>, <D,[0,0,1,1]>, <A,[1,1,1,1]>, <B,[0,0,0,1]>, <C,[0,1,1,1]>, <D,[1,1,1,1]>, <A, [0,0,1,1]>, <B,[1,1,1,1]>, <D,[0,1,1,1]>, <A, [0,0,0,1]>, <C,[0,1,1,1]>
Red()
Red()
<A, 0.542>
<B, 0.292>
<C, 0.25>
<D, 0.375>

Figure 1: Example of first phase of BDFARE algorithms applied to the fuzzy items in Table 1

line 18 of Algorithm 1). The input of this FlatMap() is a chunk of the dataset transformed into bit-lists and the variable *Global_FreqItemset* containing the frequent itemsets of length $k$. Its output will be a list of pairs comprised of the key of the itemset and its corresponding *bit_list* for each transaction. The way to perform this task is to go through all the itemsets in *Global_FreqItemset* and search in the transaction each one of the items contained in the itemset (see lines 5-7 Algorithm 3). This returns a list with the obtained itemsets and their corresponding bit-list (where the $AND$ operation has been applied to the bit-lists (see lines 8-10 Algorithm 3).

Afterwards, these pairs will be grouped by means of a function *ReduceByKey* which calculates the cardinal of the bit-lists. This function will return a list with the items and their cardinal with which we will calculate the support of each one (see lines 19-21 of Algorithm 1). Then in line 21 the *DicFreqItemset* variable is overwritten with the new candidates and the new frequent itemsets are added to the *Candidate* variable and if there is more than one new candidate a new iteration is made. These tasks are repeated until no new itemsets of length $k$ can be found (see line 26).

As mentioned in Section 1, one of the main differences with other crisp approaches developed in Spark is the use of a hash table to accelerate the searching

**Algorithm 3** Pseudocode of BitListComputation function

```
Input: Data: a fuzzy transaction $t_r$.
Input: Global_FreqItemset: List of $k$-Itemsets
Output: ItemsetList: List of pairs $< itemset, bit\_list >$.
                         BitListComputation()
ItemsetList=[ ]
For every $k$-itemset $A \in$ Global_FreqItemset
    $\{it_1, \ldots, it_m\}$ = Split($A$)
    ArrayOfBitList: $\{bit\_list_1, \ldots, bit\_list_m\}$ =SeekItems($\{it_1, \ldots, it_m\}, t_r$)
    $bit\_list_A \leftarrow bit\_list_1 \wedge \cdots \wedge \cdots \wedge bit\_list_m$
    ItemsetList.append($< A, bit\_list_A >$)
end for
return ItemsetList
```

of itemsets. When other structures, like a linear search at each node, are employed, it results in an increase of time. A comparative study among these structures can be found in [27] where the experiments confirmed that the hash table outperformed among the other data structures considered (hash tree, trie and hash table) for real and synthetic datasets.

### *3.1.4. Fuzzy Association Rule mining*

Once frequent itemsets are extracted, the final step is to uncover the fuzzy association rules that exceed the predetermined thresholds for support and confidence. This function is described in Algorithm 4.

**Algorithm 4** Main Spark procedure for extracting fuzzy association rules in BDFARE type algorithms

```
Input: Candidate: Candidate list of frequent itemsets.
Input: MinConf minimum threshold for confidence.
Output: Fuzzy Association Rules exceeding $MinConf$
DCS in $r$ chunks of Candidate: $\{C_1, \ldots, C_r\}$
    $\{< Rule_1, Conf_1 >, \ldots, < Rule_s, Conf_s >\} \leftarrow$
    $C_j$.FlatMap(GenerateRules(), Conf()) # The FlatMap generate the
    rules using the list of candidates and computes their confidence using
    the frequency information in $Global\_FreqItemset$ variable
    $Rules \leftarrow$ ReduceByKey($RuleConf \geq MinConf$)
    return Rules
end DCS computation
```

The result of previous phases is a list of frequent itemsets that will be in RDD format used by Spark. The idea of this last part of the algorithm is to use a $FlatMap$ function and afterwords a *Reduce* function as follows. For each partition of frequent itemsets the $FlatMap$ function generates the possible rules (see Figure 2 and lines 4-5 of pseudocode of Algorithm 4). This task is performed by the *GenerateRules*() function which generates rules from an itemset sequentially [50]. Therefore the distributed execution of *GenerateRules* through $FlatMap$ returns pairs of the form $< Rule, Confidence >$, and finally, through a *ReduceByKey* operation we filter those pairs keeping only those above $MinConf$ threshold (see lines 6-7 of Algorithm 4).

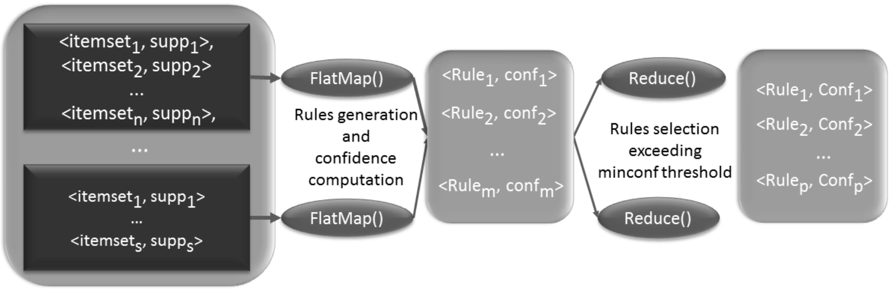


Figure 2: General association rules mining procedure using MapReduce

### 3.2. BDFARE-Apriori-TID

Another developed algorithm is BDFARE-Apriori-TID, design following the philosophy of Apriori-TID. In this case, the structure similar to the BDFARE-Apriori algorithm explained in the previous section.

#### 3.2.1. Preprocesing

The preprocessing phase is equal to the BDFARE-Apriori algorithm, because the structure of data for distributive processing across the cluster is the same.

**Algorithm 5** Main Spark procedure for BDFARE-Apriori-TID algorithm

```
Input: Data:  Fuzzy RDD transactions: $\{t_1, \ldots, t_n\}$
Input: AlphaCuts: List of alpha cuts: $\{\alpha_1, \ldots, \alpha_p\}$
Input: MinSupp: minimum support threshold.
Output: Frequent itemsets exceeding $MinSupp$ (FreqItemset)
                              Preprocessing
\\ Equal to preprocessing in Algorithm 1
                          Phase 1: FreqItems()
DCS in $q$ chunks of BitArray: $\{BA_1, \ldots, BA_q\}$
    $\{< it_1, bit\_list_1 >, \ldots, < it_m, bit\_list_m >\} \leftarrow BA_j$.FlatMap()
    $\{< it_1, card_1 >, \ldots, < it_m, card_m >\} \leftarrow$ ReduceByKey($|bit\_list_{it_k}|$)
    ItemSupport $\leftarrow$ Support($it_k$)
    DicFreqItemset $\leftarrow$ Filter($ItemSupport \geq MinSupp$)
    $BitArray \leftarrow BA_j$.Map(Remove(DicFreqItemset)).
                    .Map(Sort(DicFreqItemset)).collect()
                       Phase 2: Candidate generation
\\ Equal to lines 12-16 of Algorithm 1
do
    DCS in $q$ chunks of BitArray: $\{BA_1, \ldots, BA_q\}$
        $\{< itemset_1, bit\_list_1 >, \ldots, < itemset_m, bit\_list_m >\} \leftarrow$
        $BA_j$.FlatMap($BitListComputation(Global\_FreqItemset)$)
        # see Algorithm 3
        $\{< itemset_1, card_1 >, \ldots, < itemset_t, card_t >\} \leftarrow$
        ReduceByKey($|bit\_list_{itemset_k}|$)
        ItemsetSupport $\leftarrow$ Support($itemset_k$)
        $DicFreqItemset =$ Filter($ItemsetSupport \geq MinSupp$)
        $BitArray \leftarrow BA_j$.Map(Remove(DicFreqItemset))
    end DCS computation
    $Length++$
    $Candidate \leftarrow CandidateGen(DicFreqItemset, Length)$
    $Global\_FreqItemset.append(Candidate)$
while $|DicFreqItemset| > 1$
return $Candidate$
```

#### *3.2.2. Phase 1*

The first phase of Algorithm 5 is similar to Algorithm 1. The main difference is that those items with support value lower than the $MinSupp$ threshold are removed from the bit-list vector storage in the *Bit-Array* and sorted after the counting (see line 11). The ordering followed is from the highest to the lowest according to the number of occurrences of the item in the database.

#### *3.2.3. Phase 2*

The principal difference with BDFARE-Apriori is that in each iteration of the frequent itemsets searching process all infrequent items are removed (see line 22 Algorithm 5).

#### *3.2.4. Fuzzy Association Rule mining*

The extraction rules phase is equal to the BDFARE-Apriori algorithm explained in Section 3.1.4. We use the hash table obtained with the frequent itemsets for extracting the association rules and calculate the measures like the confidence, lift or certainty factor of each rule.

### *3.3. BDFARE-ECLAT*

We have also implemented the BDFARE algorithm following the philosophy of the sequential ECLAT algorithm, but using the MapReduce paradigm in the Spark framework. In this case, the principal difference resides in the data distribution by itemset, instead of process distribution by set of transactions like in Apriori.

#### *3.3.1. Preprocesing*

The preprocessing in BDFARE-ECLAT algorithm is different from the previous explained algorithms. In this way, BDFARE-ECLAT needs the data grouped

**Algorithm 6** Main Spark procedure for BDFARE-ECLAT algorithm

```
Input: Data:  Fuzzy RDD transactions: $\{t_1, \ldots, t_n\}$
Input: AlphaCuts: List of alpha cuts: $\{\alpha_1, \ldots, \alpha_p\}$
Input: MinSupp: Minimum support threshold.
Output: Frequent itemsets exceeding $MinSupp$
                         Preprocessing
DCS in $q$ chunks of Data: $\{S_1, \ldots, S_q\}$
    $BitArray_{S_i} \leftarrow S_i$.Map(FuzzyToArray($t_k \in S_i$)).GroupByKey()
    # Map function computes independently each transaction in $S_i$
    # $BitArray_{S_i}$ contains lists of the form $< item,$
    $[bit\_list_1, \ldots, bit\_list_n] >$, in this way, they can be distributed in the
    cluster by item (see Figure 3)
                         Phase 1: FreqItems()
DCS in $q$ chunks of BitArray: $\{BA_1, \ldots, BA_q\}$
    $\{< it_1, [bit\_list_1, \ldots, bit\_list_n] >, \ldots, < it_m, [bit\_list_1, \ldots, bit\_list_n] >\} \leftarrow$
    $BA_j$.FlatMap()
    $\{< it_1, card_1 >, \ldots, < it_m, card_m >\} \leftarrow$ ReduceByKey($|bit\_list_{it_k}|$)
    ItemSupport $\leftarrow$ Support($it_k$)
    DicFreqItemset $\leftarrow$ Filter($ItemSupport \geq MinSupp$)
                         Phase 2: Candidate generation
$Candidate \leftarrow DicFreqItemset$ #Candidates of $Length = 1$
$Length = 2$
$Global\_FreqItemset \leftarrow Candidate$
$broadcast(Global\_FreqItemset)$ #Creates a broadcast variable for using across the cluster
do
    DCS in $q$ chunks of items list of BitArray: $\{IBA_1, \ldots, IBA_q\}$
        $\{< itemset_1, bit\_list_1 >, \ldots, < itemset_m, bit\_list_m >\} \leftarrow$
        $IBA_j$.FlatMap($BitListComputation(Global\_FreqItemset)$)
        # see Algorithm 3
        $\{< itemset_1, card_1 >, \ldots, < itemset_t, card_t >\} \leftarrow$
        ReduceByKey($|bit\_list_{itemset_k}|$)
        ItemsetSupport $\leftarrow$ Support($itemset_k$)
        $DicFreqItemset =$ Filter($ItemsetSupport \geq MinSupp$)
    end DCS computation
    $Length++$
    $Candidate \leftarrow CandidateGen(DicFreqItemset, Length)$
    $Global\_FreqItemset.append(Candidate)$
while $|DicFreqItemset| > 1$
return $Candidate$
```

by item joint with their membership degree (in the format of bit-list) (see Algorithm 6). Therefore the main difference is the aggregation by item in the MapReduce phase depicted in Figure 3, where there is an example of the aggregation function and output data of BDFARE-ECLAT preprocessing.

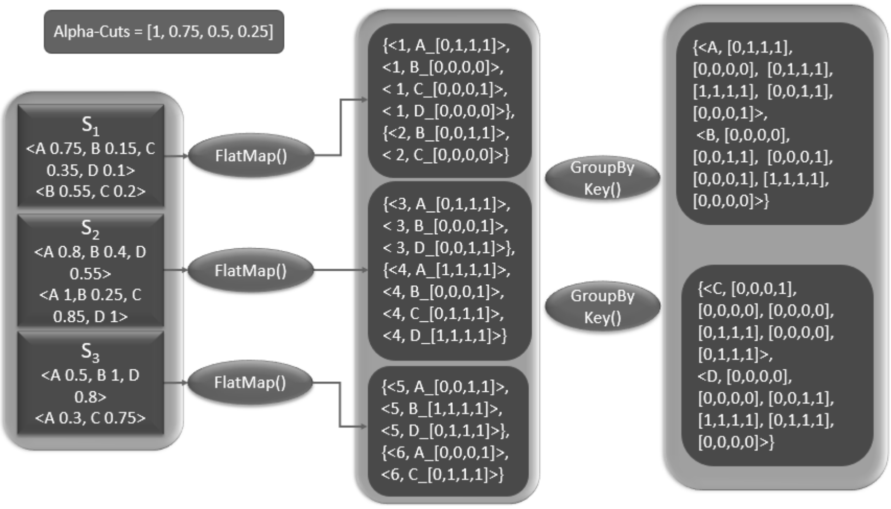

Alpha-Cuts = [1, 0.75, 0.5, 0.25]
S1
<A 0.75, B 0.15, C 0.35, D 0.1>
<B 0.55, C 0.2>
S2
<A 0.8, B 0.4, D 0.55>
<A 1,B 0.25, C 0.85, D 1>
S3
<A 0.5, B 1, D 0.8>
<A 0.3, C 0.75>
FlatMap()
FlatMap()
FlatMap()
{<1, A_[0,1,1,1]>, <1, B_[0,0,0,0]>, < 1, C_[0,0,0,1]>, < 1, D_[0,0,0,0]>}, {<2, B_[0,0,1,1]>, < 2, C_[0,0,0,0]>}
{<3, A_[0,1,1,1]>, < 3, B_[0,0,0,1]>, < 3, D_[0,0,1,1]>}, {<4, A_[1,1,1,1]>, <4, B_[0,0,0,1]>, <4, C_[0,1,1,1]>, <4, D_[1,1,1,1]>}
{<5, A_[0,0,1,1]>, <5, B_[1,1,1,1]>, <5, D_[0,1,1,1]>}, {<6, A_[0,0,0,1]>, <6, C_[0,1,1,1]>}
GroupBy Key()
GroupBy Key()
{<A, [0,1,1,1], [0,0,0,0], [0,1,1,1], [1,1,1,1], [0,0,1,1], [0,0,0,1]>, <B, [0,0,0,0], [0,0,1,1], [0,0,0,1], [0,0,0,1], [1,1,1,1], [0,0,0,0]>}
{<C, [0,0,0,1], [0,0,0,0], [0,0,0,0], [0,1,1,1], [0,0,0,0], [0,1,1,1]>, <D, [0,0,0,0], [0,0,0,0], [0,0,1,1], [1,1,1,1], [0,1,1,1], [0,0,0,0]>}

Figure 3: Preprocessing data phase which transforms fuzzy items into bit-lists in BDFARE-ECLAT

Firstly, the algorithm transforms each transaction into pairs $< item_i, bit\_list_i >$ as explained in the preprocesing phase of BDFARE-Apriori algorithm. Then, the algorithm aggregates the different lists by items using a *GroupByKey* function (see line 6 of Algorithm 6), generating pairs with an item and a large list of lists containing a bit-list per transaction.

### *3.3.2. Phase 1*

In the first phase, the algorithm counts the number of appearances of every item in every transaction depending on the established set of $\alpha$-cuts. After that, the algorithm calculates frequent items and generates the itemsets for the next step. The main difference with BDFARE-Apriori resides in how to compute the fre-

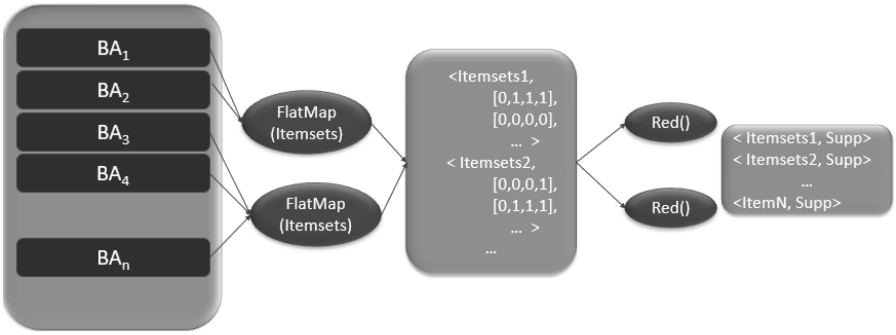


Figure 4: Example of second phase of BDFARE-ECLAT applied to the fuzzy items in Table 2

quency of items using the list data format. In this case the *FlatMap* function (see line 8 of Algorithm 6) return pairs comprised of items and their corresponding *bit-lists* per transaction. Then, the *ReduceByKey* function calculates the cardinal per item with this data, and afterwards the support is computed (see lines 9-11 of Algorithm 6).

### *3.3.3. Phase 2*

For the calculation of itemsets support, the algorithm uses the *FlatMap*() function in the same way as in the Apriori case. The main difference is how the pair lists are computed. In this case, the *FlatMap*() function returns a pair comprised of an itemset and a list for each transaction with the *bit-list* using the *numpy* array format (see Figure 4 and lines 18-19 of Algorithm 6). Therefore, the algorithm calculates all the cardinalities needed for the computation of the cardinality of the obtained itemsets using the *ReduceByKey*() function, and afterwards computes their support. This step will be repeated until no new $k$-itemsets can be found.

### *3.3.4. Fuzzy Association Rule mining*

The extraction rules phase is equal to BDFARE-Apriori and BDFARE-Apriori-

TID previously explained.

## 4. Experiments and results

Our aim is to study the behaviour of the new algorithms designed for distributed computation using Spark framework. The experiments carried out have been designed to analyze the following different aspects:

- Compare proposed algorithms among them taking into account the execution time along different configurations for the experiments, and towards different datasets.
- Compare the proposed algorithms for the calculation of measures like support and confidence when the algorithms use *numpy* vector or traditional method.
- Analyse proposed algorithms measuring the speed up and the efficiency achieved by increasing the number of cores in the procedure.
- Study the performance of proposed algorithms with respect to different sets of $\alpha$-cuts.

To this end, the algorithms have been tested in several fuzzy transactional datasets to analyse their running time attending to different parameters: the number of items, the length of datasets, the number of transactions, and the number of $\alpha$-cuts.

Five different datasets have been considered. Three of them from the UCI machine learning repository[3] where some continuous attributes have been conveniently fuzzyfied as described in [45]. The `Bank`[4] dataset contains marketing data from different campaigns of a Portuguese banking institution. The continuous attributes of this dataset have been fuzzified by defining suitable fuzzy partitions according to the semantics of the attributes. The `Higgs` [51] dataset

[3] http://archive.ics.uci.edu/ml/

[4] http://ugritlab.ugr.es:82/cjferba/datasets

is comprised of several attributes about Higgs Boson produced during several Monte Carlo simulations. It is contains 28 continuous attributes plus the class attribute. The continuous attributes have been fuzzified using the following linguistic labels: low, medium and high as it can be seen in Figure 7, and the class attribute remains the same containing the values 1 or 0.

The `Forest-equidepth` database is originated from the database used in [52] where binary attributes were excluded and we have fuzzified the remaining attributes using equi-depth intervals explained in [53]. The `Energy ICPE` dataset, comprises data collected from an office building in Romania located in Bucharest. It comprises 273 sensors containing different metering data collected from September 2016 to September 2017 with a total of 3,649,678 transactions. Continuous attributes in this dataset have been fuzzified by the process described in [54]. Some of the datasets used in the experiments are replicated to prove the performance of the algorithm in extreme situations. In Table 2 it can be also found their original size between parentheses.

| Fuzzy Database | Transactions | Fuzzy Items |
|---|---:|---:|
| `Bank` | 1446752(45211) | 112 |
| `Higgs` | 11000000 | 86 |
| `Forest-equidepth` | 581012 | 37 |
| `Energy ICPE` | 3649678 | 1121 |

Table 2: Datasets

The experimental evaluation has been made in a cluster consisting of 4 servers with 102 cores and 420 Gb of RAM where intel's hyperthreading functionality was disabled for testing. The Spark version employed was 2.2 which uses a fully distributed mode with Ambari Server.

We have made several experiments with different threshold values $Minsupp \in \{0.2, 0.4\}$ and $Minconf \in \{0.8\}$. The high values fixed for the minimum support have been chosen to reduce the number of obtained rules, because some of the fuzzy items are very frequent in the databases, leading to a huge amount of discovered rules.

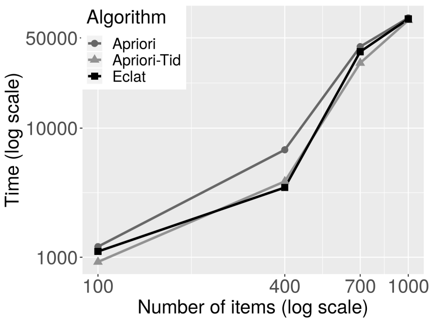


Figure 5: Performance in logarithmic scale of BDFARE algorithms for `Energy ICPE` dataset when the quantity of items increases

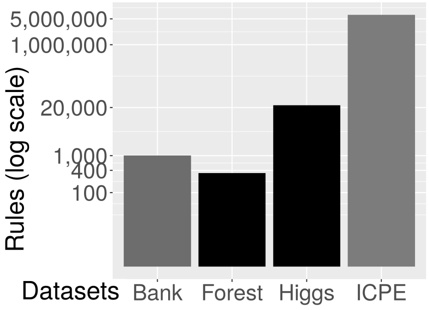


Figure 6: Number of rules extracted for each dataset with $MinSupp = 0.2$ for all datasets except `Energy ICPE` ($MinSupp = 0.4$) and $MinConf = 0.8$

Figure 6 shows the number of rules obtained for a minimum support equal to 0.2 for all datasets except for the case of `Energy ICPE` ($MinSupp = 0.4$) and minimum confidence values were set to 0.8. We are interested in observing the performance (time) of all approaches depending on the number of transactions and the number of items and the set of $\alpha$-cuts. In addition to this, note that the proposed algorithms extract rules without restricting the number of items appearing in the consequent or the antecedent of the rules.

We began by analysing the behaviour of the algorithms when the number of

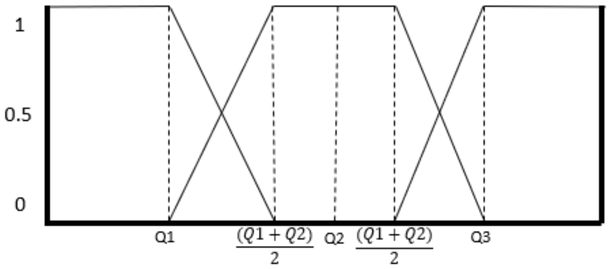


Figure 7: Fuzzyfication of Higgs features using quartiles

resources (cores and memory) increases. Figures 8, 9, 10 and 11 shows the behaviour of the algorithms when the resources (1 core means sequential and 24, 48, 72, 102 is equivalent to 1,2,3,4 nodes where each node have 100 Gb of RAM) increases in each of the datasets. It can be observed that BDFARE-Apriori performed better than BDFARE-ECLAT. Moreover, BDFARE-Apriori-TID is more efficient than the BDFARE-Apriori, as expected.

And BDFARE-ECLAT has some memory problems in `Higgs` and `Energy ICPE` dataset, because these datasets have a lot of transactions and items and the lists that BDFARE-ECLAT has to create are very big. For this reason it needs to use a big amount of memory. To be exhaustive, in those datasets with 1 million and 11 million transactions, the algorithms achieved in average time savings of 40% and 60% respectively if they are compared to the non-distributed case (1 core). This graph also shows that BDFARE-ECLAT did not complete its execution for `Higgs` dataset, which contains 11 million of transactions, and `Energy ICPE` dataset, containing 1000 items, due to a memory overflow error, i.e due to the lack of memory. By contrast, BDFARE-Apriori and BDFARE-Apriori-TID algorithms completed the execution.

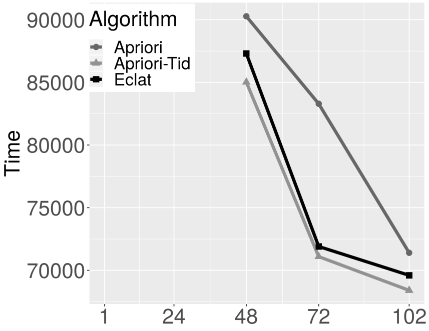


Figure 8: Time for different number of clusters for `Energy ICPE` dataset

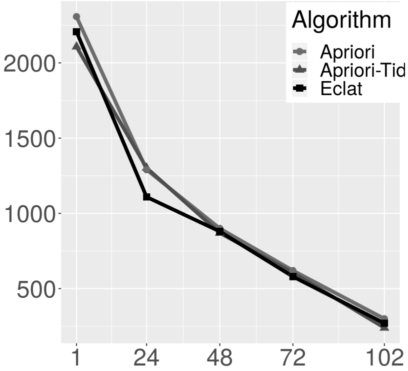
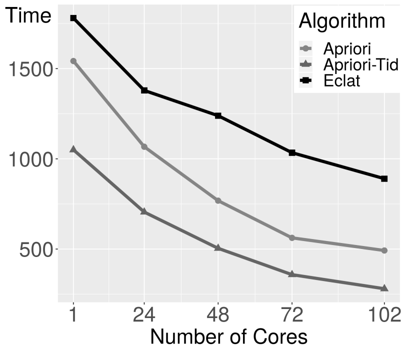


Figure 9: Time for different number of clusters for `Forest-equidepth` dataset

Figure 5 shows the time spent by the algorithms when the number of items



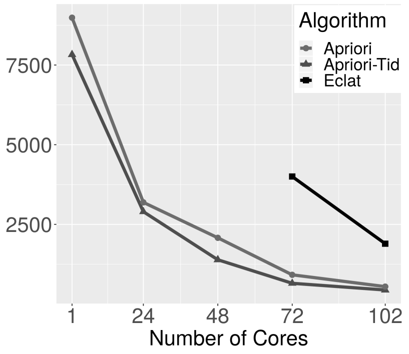
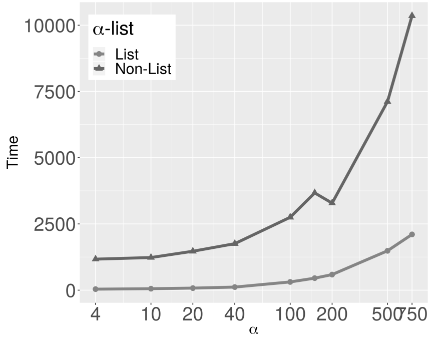


Figure 10: Time for different number of clusters for `Bank` dataset

Figure 11: Time for different number of clusters for `Higgs` dataset



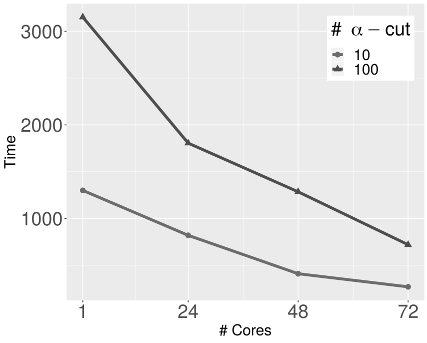


Figure 12: Performance in logarithmic scale of BDFARE-Apriori-TID algorithm when the quantity of $\alpha$-cuts increases in `forest-equidepth` dataset

Figure 13: Performance in logarithmic scale of BDFARE-Apriori-TID algorithm with 10 and 100 $\alpha$-cuts when the quantity of cores increases (from 1 core=sequential to 32 cores) in `Bank` dataset

575 increases for `Energy ICPE` dataset. In this graph it can be observed that BDFARE does not offer substantial improvements in terms of time, because the increment of time is exponential by the number of items. This is due to the items combinatorial explosion of Apriori-based algorithms. Additionally, BDFARE does not achieve always more efficient executions, because during the

jobs planning, necessary when distributing data, there is also a waste of time. However the performance of the BDFARE algorithm tends to improve when the number of transactions increases. This is because the distribution of data across the clusters is made by transaction. Moreover we can see that the BDFARE-ECLAT algorithm, although it uses a list of items, the performance is similar to BDFARE-Apriori. This is because the use of long lists of elements in each node does not work efficiently.

For the analysis of the *speed up* and the *efficiency* [55, 56, 57] according to the number of cores, the known measure of speed up has been employed, defined as [57, 58]

$$S_n = T_1/T_n \tag{10}$$

where $T_1$ represents the time of the sequential algorithm and $T_n$ the time of the distributed algorithm with several cores. The efficiency measure [55, 56, 57] is defined as

$$E_n = S_n/n = T_1/(n \cdot T_n) \tag{11}$$

In Figures 14 and 15 it can be seen that as the numbers of cores increases, the efficiency and speed up are improved, even they are not optimal. This factor is influenced by the cores workloads and the network congestion employed for the communication among the cores.

In addition, Figure 14 shows the evolution of the execution times (speed up) for the different proposals. In this figure, it can be observed that the greatest reduction in time execution is achieved when the number of processors is higher. Although the speed up obtained increased along the number of processors used, we can see that it moves away from proportional speed up as resources expand. This is because increased resources may not be used as efficiently with respect to the same amount of data using fewer resources. Note also that there are parts of the algorithm that are iterative (e.g. calculation of the itemset of size $k+1$ needs to execute those of length $k$), so certain parts are not totally distributed and this actually affects efficiency. As far as we can see in Figure 15 the efficiency does not increase proportionally. This same behavior of efficiency and speed up has been observed in other studies of distributed algorithms where the efficiency does not increase proportionality with more processors [59, 57].

Regarding the performance of BDFARE algorithms with respect to the set of $\alpha$-cuts we have performed different experiments. Figure 12 describes how BDFARE-Apriori-TID algorithm performs when it uses lists to represent $\alpha$-cuts and the number of $\alpha$-cuts is increased, using from 1 to 16 cores (1, 4, 8, 16 cores). The difference of the performance when the algorithm uses lists or not for different number of $\alpha$-cuts is noticeable, concluding that using lists improves the efficiency.

Moreover, we have observed that the number of rules found considering 100, 50 $\alpha$-cuts and 10 $\alpha$-cuts is the same (in all the experiments). Then, it seems that using only 10 $\alpha$-cuts it is sufficient to achieve the precision of computations to retrieve all the fuzzy association rules, which will decrease the number of computations and therefore, the efficiency of the BDFARE algorithms.

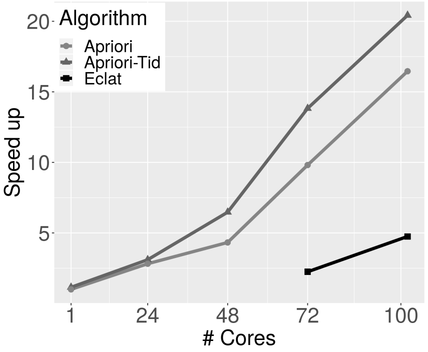


Figure 14: Speed up of BDFARE-Apriori-TID algorithm for `Higgs` dataset

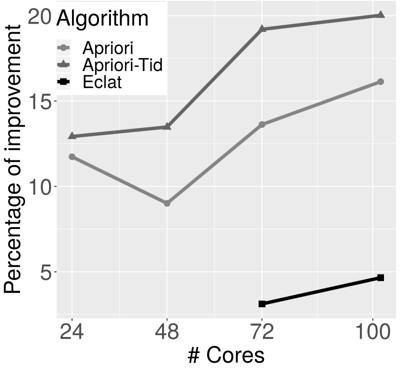


Figure 15: Efficiency of BDFARE-Apriori-TID algorithm for `Higgs` dataset measured by percentage of improvement

## 5. Conclusions and future research

This paper has proposed different fuzzy association rule mining algorithms in the ambit of Big Data, which allows to extract co-occurrence patterns from fuzzy datasets. It has been shown, that non-distributed algorithms proposed for mining fuzzy association rules can fail when analysing massive datasets due to the memory overflow errors and their efficiency is affected when the dataset grows.

To this end, the proposals presented for mining fuzzy association rules using the Spark framework are capable of analysing massive data. For that, the different proposals have been compared and analysed obtaining improvements not only in the execution time, but also in their memory, improving their processing capacity (note that some of the experiments could not finish their execution in the non-distributed cases). An additional advantage of Big Data proposals is that their performance can be easily improved even further just by expanding the system by adding more clusters (computation nodes). This makes easier to scale our proposals to be executed in external data centers or in cloud systems such as AWS (Amazon Web Services).

Additionally, our proposal is based on a decomposition of interestingness measures in terms of $\alpha$-cuts which facilitates their implementation to other interestingness measures different to that of support and confidence using the formal model developed in [12], and we have experimentally demonstrated that it is sufficient to consider only 10 equidistributed $\alpha$-cuts in order to mine all significant fuzzy association rules.

As regards future research, we plan the application of these proposals to real world problems such as in the analysis of social media or in the energy field, by the combination of other Data Mining and Machine Learning techniques in order to obtain more valuable knowledge from the data, utilising association rules as a first exploratory step in the knowledge discovery process.

Lastly, we also intend to generalise these Big Data procedures to other techniques that are formulated in terms of association rules such as gradual dependencies [60], [61], exception and anomalous rules [62], etc.

*Acknowledgements*

The research reported in this paper was partially supported the COPKIT project from the 8$^{\text{th}}$ Programme Framework (H2020) research and innovation programme (grant agreement No 786687) and the Spanish Ministry for Economy and Competitiveness by the project grant TIN2015-64776-C3-1-R.

## References

[1] S. del Río, V. López, J. M. Benítez, F. Herrera, On the use of mapreduce for imbalanced big data using random forest, Information Sciences 285 (2014) 112 – 137, processing and Mining Complex Data Streams.

[2] D. Anastasiu, J. Iverson, S. Smith, G. Karypis, Frequent Pattern Mining, Springer International Publishing Switzerland, 2014, Ch. Big Data Frequent Pattern Mining, pp. 225–259.

[3] A. Fernández, C. Carmona, M. del Jesus, F. Herrera, A view on fuzzy systems for big data: Progress and opportunities, International Journal of Computational Intelligence Systems 9 (2016) 69–80.

[4] C. Fernandez-Basso, M. Ruiz, M. Martin-Bautista, Extraction of association rules using big data technologies, Int. J. of Design & Nature and Ecodynamics 11 (3) (2016) 178–185.

[5] E. Hüllermeier, Y. Yi, In defense of fuzzy association analysis, IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics 37 (4) (2007) 1039–1043.

[6] L. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.

[7] M. Delgado, N. Marín, D. Sánchez, M. Vila, Fuzzy association rules: General model and applications, IEEE Transactions on Fuzzy Systems 11 (2) (2003) 214–225.

[8] J. Calero, G. Delgado, M. Sánchez-Marañón, D. Sánchez, J. Serrano, M. A. V. Miranda, Helping user to discover association rules: A case in soil color as aggregation of other soil properties, in: ICEIS 2003, Proceedings of the 5th International Conference on Enterprise Information Systems, Angers, France, April 22-26, 2003, 2003, pp. 533–540.

[9] X. M. et al., Mllib: Machine learning in apache spark, Journal of Machine Learning Research 17 (2016) 1–7.
URL `http://arxiv.org/abs/1505.06807`

[10] T. White, Hadoop: The Definitive Guide. Fourth edition, O'Reilly, 2015.

[11] L. Liu, Performance comparison by running benchmarks on hadoop, spark and hamr, Ph.D. thesis, University of Delaware (2016).
URL `http://udspace.udel.edu/bitstream/handle/19716/17628/2015_LiuLu_MS.pdf?sequence=1`

[12] M. Delgado, M. D. Ruiz, D. Sánchez, J.-M. Serrano, A formal model for mining fuzzy rules using the rl representation theory, Information Sciences 181 (23) (2011) 5194–5213.

[13] R. Agrawal, R. Srikant, Fast Algorithms for Mining Association Rules in Large Databases, in: Proc. of the Twentieth Inter. Conf. on Very Large Databases, Santiago, Chile, 1994, pp. 487–499.

[14] J. Hipp, U. Güntzer, G. Nakhaeizadeh, Algorithms for association rule mining - a general survey and comparison, ACM sigkdd explorations newsletter 2 (1) (2000) 58–64.

[15] M. J. Zaki, S. Parthasarathy, M. Ogihara, W. Li, et al., New algorithms for fast discovery of association rules., in: KDD, Vol. 97, 1997, pp. 283–286.

[16] M. J. Zaki, Scalable algorithms for association mining, IEEE Transactions on Knowledge and Data Engineering 12 (3) (2000) 372–390.

[17] C. Borgelt, Efficient implementations of apriori and eclat, in: FIMI'03: Proc. of the IEEE ICDM workshop on frequent itemset mining implementations, 2003.

[18] C. R. J. Li, Z. H. Deng, Mining frequent ordered patterns without candidate generation, in: in FSKD 2007, ACM Press, New York, New York, USA, 2007, pp. 402–406. `arXiv:0611061v2`, `doi:10.1109/FSKD.2007.402`.
URL `http://portal.acm.org/citation.cfm?doid=342009.335372`

[19] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, in: SIGMOD'00 Proc. of the 2000 ACM SIGMOD Int. conf. on Management of data, 2000, pp. 1–12.

[20] Z. Zheng, R. Kohavi, L. Mason, Real world performance of association rule algorithms, in: Proc. of the seventh ACM SIGKDD Int. conf. on Knowledge discovery and data mining, ACM, 2001, pp. 401–406.

[21] K. Garg, D. Kumar, Comparing the performance of frequent pattern mining algorithms, International Journal of Computer Applications 69 (25).

[22] C. Fernandez-Basso, M. Ruiz, M. Martin-Bautista, A comparative analysis of spark frequent itemsets and association rule mining algorithms, Submitted to Knoledge-Based Systems.

[23] H. Li, Y. Wang, D. Zhang, M. Zhang, E. Y. Chang, PFP: parallel fp-growth for query recommendation, in: Proc. of the 2008 ACM conference on Recommender systems, ACM, 2008, pp. 107–114.

[24] N. Li, L. Zeng, Q. He, Z. Shi, Parallel implementation of apriori algorithm based on mapreduce, in: Proc. of the 2012 13th ACIS Int. conf. on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD '12, IEEE Computer Society, Washington, DC, USA, 2012, pp. 236–241.

[25] Z. Farzanyar, N. Cercone, Efficient mining of frequent itemsets in social network data based on mapreduce framework, in: Proc. in ASONAM 2013, 2013, pp. 1183–1188.

[26] Z. Farzanyar, N. Cercone, Accelerating frequent itemset mining on the cloud: A mapreduce-based approach, in: IEEE 13th Int. Conf. on Data Mining Workshops, 2013, pp. 592–598.

[27] S. Singh, R. Garg, P. Mishra, Performance analysis of apriori algorithm with different data structures on hadoop cluster, International Journal of Computer Applications 128 (9) (2015) 45–51.

[28] S. Rathee, M. Kaul, A. Kashyap, R-apriori: An efficient apriori based algorithm on spark, in: Proc. of the PIKM'15, ACM, Melbourne, VIC, Australia, 2015.

[29] H. Qiu, R. Gu, C. Yuan, Y. Huang, Yafim: A parallel frequent itemset mining algorithm with spark, in: Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International, IEEE, 2014, pp. 1664–1671.

[30] R. Agrawal, J. C. Shafer, Parallel mining of association rules, IEEE Transactions on knowledge and Data Engineering 8 (6) (1996) 962–969.

[31] M. J. Zaki, S. Parthasarathy, M. Ogihara, W. Li, Parallel algorithms for discovery of association rules, Data mining and knowledge discovery 1 (4) (1997) 343–373.

[32] S. Cong, J. Han, J. Hoeflinger, D. Padua, A sampling-based framework for parallel data mining, in: Proc. of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming, ACM, 2005, pp. 255–265.

[33] T. Shintani, M. Kitsuregawa, Hash based parallel algorithms for mining association rules, in: Parallel and Distributed Information Systems, 1996., Fourth Int. conf. on, IEEE, 1996, pp. 19–30.

[34] A. Mueller, Fast sequential and parallel algorithms for association rule mining: A comparison, Tech. rep., University of Maryland at College Park College Park, MD, USA (1998).

[35] M. Gabroveanu, M. Cosulschi, F. Slabu, Mining fuzzy association rules using mapreduce technique, in: Int. Symposium on INnovations in Intelligent SysTems and Applications, INISTA, 2016, pp. 1–8.

[36] M. Gabroveanu, I. Iancu, M. Cosulschi, N. Constantinescu, Towards using grid services for mining fuzzy association rules, in: Proc. of the 1st East European Workshop on Rule-Based Applications, RuleApps, 2007, pp. 507–513.

[37] M. Gabroveanu, M. Cosulschi, N. Constantinescu, A new approach to mining fuzzy association rules from distributed databases, Annals of the University of Bucharest LIV (2005) 3–16.

[38] R. Agrawal, T. Imielinski, A. Swami, Mining associations between sets of items in large databases, in: ACM-SIGMOD Int. Conf. on Data, 1993, pp. 207–216.

[39] P. Hájek, The question of a general concept of the GUHA method, Kybernetika 4 (1968) 505–515.

[40] P. Hájek, T. Havranek, Mechanizing Hypothesis Formation, Springer Verlag: Berlin, 1978.

[41] J. Calero, G. Delgado, M. Sánchez-Marañón, D. Sánchez, M. A. V. Miranda, J. Serrano, An experience in management of imprecise soil databases by means of fuzzy association rules and fuzzy approximate dependencies, in: ICEIS 2004, Porto, Portugal, April 14-17, 2004, 2004, pp. 138–146.

[42] F. Berzal, M. Delgado, D. Sánchez, M. Vila, Measuring accuracy and interest of association rules: A new framework, Intelligent Data Analysis 6 (3) (2002) 221–235.

[43] N. Marín, M. Ruiz, D. Sánchez, Fuzzy frameworks for mining data associations: fuzzy association rules and beyond, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 6 (2) (2016) 50–69.

[44] M. Delgado, M. Ruiz, D. Sánchez, J. Serrano, A formal model for mining fuzzy rules using the RL representation theory, Information Sciences 181 (23) (2011) 5194–5213.

[45] M. D. Ruiz, D. Sánchez, M. Delgado, M. J. Martin-Bautista, Discovering fuzzy exception and anomalous rules, IEEE Transactions on Fuzzy Systems 24 (4) (2016) 930–944.

[46] D. Dubois, E. Hüllermeier, H. Prade, A systematic approach to the assessment of fuzzy association rules, Data Mining and Knowledge Discovery 13 (2) (2006) 167–192.

[47] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, I. Stoica, Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing, in: Proc. of the 9th USENIX conference on Networked Systems Design and Implementation, USENIX Association, 2012, pp. 2–2.

[48] E. Louie, T. Young, Finding association rules using fast bit computation: Machine-oriented modeling, in: International Symposium on Methodologies for Intelligent Systems, Springer, 2000, pp. 486–494.

[49] J. Rauch, M. Šimůnek, Foundations of Data Mining and knowledge Discovery, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, Ch. An Alternative Approach to Mining Association Rules, pp. 211–231. `doi:10.1007/11498186_13`.
URL `https://doi.org/10.1007/11498186_13`

[50] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. I. Verkamo, et al., Fast discovery of association rules., Advances in knowledge discovery and data mining 12 (1) (1996) 307–328.

[51] P. Baldi, P. Sadowski, D. Whiteson, Searching for exotic particles in high-energy physics with deep learning, Nature communications 5 (2014) 4308.

[52] H. Liu, F. Hussain, C. L. Tan, M. Dash, Discretization: An enabling technique, Data mining and knowledge discovery 6 (4) (2002) 393–423.

[53] M. Calvo-Flores, M. Ruiz, D. Sánchez, J. Serrano, A fuzzy rule mining approach involving absent items, in: Proc. of the EUSFLAT'2011, 2011, pp. 275 – 282. doi:10.2991/eusflat.2011.126.

[54] C. Fernandez-Basso, M. Ruiz, M. Martin-Bautista, A fuzzy mining approach for energy efficiency in a big data framework, IEEE Transactions on Fuzzy Systemsdoi:10.1109/TFUZZ.2020.2992180.

[55] V. P. Kumar, A. Gupta, Analyzing scalability of parallel algorithms and architectures, Journal of parallel and distributed computing 22 (3) (1994) 379–391.

[56] A. Y. Grama, A. Gupta, V. Kumar, Isoefficiency: Measuring the scalability of parallel algorithms and architectures, IEEE Parallel & Distributed Technology: Systems & Applications 1 (3) (1993) 12–21.

[57] C. Barba-González, J. García-Nieto, A. Benítez-Hidalgo, A. J. Nebro, J. F. Aldana-Montes, Scalable inference of gene regulatory networks with the spark distributed computing platform, in: J. Del Ser, E. Osaba, M. N. Bilbao, J. J. Sanchez-Medina, M. Vecchio, X.-S. Yang (Eds.), Intelligent Distributed Computing XII, Springer International Publishing, Cham, 2018, pp. 61–70.

[58] F. J. Baldán, J. M. Benítez, Distributed fastshapelet transform: a big data time series classification algorithm, Information Sciences.

[59] C. Barba-Gonzaléz, J. García-Nieto, A. J. Nebro, J. F. Aldana-Montes, Multi-objective big data optimization with jmetal and spark, in: H. Trautmann, G. Rudolph, K. Klamroth, O. Schütze, M. Wiecek, Y. Jin,

C. Grimme (Eds.), Evolutionary Multi-Criterion Optimization, Springer International Publishing, Cham, 2017, pp. 16–30.

[60] E. Hüllermeier, Association rules for expressing gradual dependencies, in: Proc. PKDD 2002 Lecture Notes in Computer Science, 2431, 2002, pp. 200–211.

[61] F. Berzal, J. Cubero, D. Sánchez, M. Vila, An alternative approach to discover gradual dependencies, Int. Journal of Uncertainty, Fuzziness and Knowledge-based Systems 15 (5) (2007) 559–570.

[62] M. Delgado, M. Ruiz, D. Sánchez, New approaches for discovering exception and anomalous rules, Int. J. of Uncert., Fuzziness and Knowledge-Based Systems 19 (2) (2011) 361–399.

# 3 Big Data frequent itemsets mining in streaming environment

## 3.1 Finding Tendencies in Streaming Data using Big Data Frequent Itemset Mining(Published in Knowledge-Based Systems)

- Carlos Fernandez-Basso, Abel J.Francisco-Agra, M. Dolores Ruiz, Maria J. Martin-Bautista. Knowledge Based Systems (2019).
  - Status: **Published**.
  - Impact Factor (JCR 2018): **5.101**
  - Subject Category: **Computer Science, Information Systems**
  - Rank: **27/155**
  - Quartile: **Q1**

# Finding Tendencies in Streaming Data using Big Data Frequent Itemset Mining

Carlos Fernandez-Basso, Abel J. Francisco-Agra, Maria J. Martin-Bautista

*Department of Computer Science and A.I. and CITIC-UGR, University of Granada, Spain*

M. Dolores Ruiz

*Computer Engineering Department, University of Cádiz, Spain*

**Abstract**

The amount of information generated in social media channels or economical/business transactions exceeds the usual bounds of static databases and is in continuous growing. In this work, we propose a frequent itemset mining method using sliding windows capable of extracting tendencies from continuous data flows. For that aim, we develop this method using Big Data technologies, in particular, using the Spark Streaming framework enabling distributing the computation along several clusters and thus improving the algorithm speed. The experimentation carried out shows the capability of our proposal and its scalability when massive amounts of data coming from streams are taken into account.

*Keywords:* Streaming data, Big Data, frequent itemset mining, tendencies

## 1. Introduction

Nowadays, there are many techniques to obtain information from data and they are related to Data Mining. Two important factors distinguish the type of technique we have to use: the type of data we have and the kind of information we want to obtain. Recently, tendency analysis is a must due to the need of constant updated information on economic fluctuations, social media tendencies, etc. In this area, the extraction of frequent items has proved useful, as shown in [1], but the continuous change of data and their enormous volume complicate their extraction with classic techniques.

Big Data has arisen as a new framework to store and process large amounts of data enabling the distribution of computations among several clusters. New techniques appear in order to store and process data such as non-structured storage frameworks as NoSQL databases [2], which allow the programmer to

*Email address:* mariadolores.ruiz@uca.es (M. Dolores Ruiz)

abstract the complexity of data management in large clusters. An example of this is the HDFS platform [3]. The main paradigm behind the Big Data used for distributed programming is known as Map-Reduce [4] and is available in platforms such as Hadoop [5] or Spark [6]. The libraries in these two platforms are getting larger in recent years thanks to the fast deployment of suitable extensions of existent algorithms within the Data Mining and Machine Learning communities. However, some of these implementations are not direct extensions, since we have to deal with the peculiarity of the algorithms and the data structures they employ. This is for instance the case of structures like trees which perform very efficiently in the non-distributive case and not so well along several clusters (we recommend reading the analysis made in [7]).

In frequent itemset mining, several sequential approaches have been traditionally used such as the Apriori (or some of its advanced versions like Apriori-TID), ECLAT, or FP-growth algorithms [8] to extract frequent co-occurrence of items in a database. But these algorithms cannot be used for continuous data flows, where time plays an essential role. The sub-field in charge of this is known as Streaming analysis. Several approaches have been developed for frequent itemset extraction in data streams, but sometimes their processing capacity is exceeded when data comes from social media or marketing fluctuations. Existent algorithms for stream mining sometimes fail when the volume of data contained in a temporal window cannot be processed due to a memory-overflow. At this respect, distributed algorithms are capable of processing the data by mapping them into different clusters of computation. In particular, Big Data implementations enable to handle with these types of problems, allowing not only their processing but also providing redundancy mechanisms to prevent the data loss without having data inconsistency. However, as far as we are concerned, there are no available Big Data implementations for frequent itemset stream mining till the moment.

In the light of these observations, this paper presents a new proposal to discover tendencies using frequent itemset mining in continuous stream data. For that, we have reviewed and analyzed existent algorithms, and we propose an improved Big Data version using the Spark Streaming library of the FIMoTS (Frequent Itemset Mining over Time-sensitive Streams) algorithm developed in [9]. It is worthwhile to stress that the proposed algorithm is not a straightforward extension of the existent FIMoTS. This is because it lays on a tree structure which increments complexity if we handle it in a distributed way. Instead, we have considered a different configuration in order to manage with the necessary information. The conducted experiments show that the new distributive proposal solves the limitations of sequential FIMoTS approach when massive streaming data are considered, outperforming the original one in all the cases analyzed and scaling very well in very large data streams.

The paper is structured as follows. Next section introduces the problem under study and reviews the existent approaches in frequent itemset mining from different perspectives. Section 3 presents our proposal for mining frequent itemsets in data streams using Big Data techniques. Section 4 shows a comparison between our proposal and its non-distributive version and we also analyze the performance of our proposal. The paper ends with the conclusions and future plans.

## 2. Preliminaries and Related Works

Since its appearance, Frequent Pattern Mining (FPM) has been extensively applied in Association rule mining as a sub-phase to obtain the frequent items occurring together in a set of transactions [10]. Formally, let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items and $D = \{t_1, t_2, \ldots, t_N\}$ be a set of $N$ transactions where each $t_j$ contains a subset of items.

The problem of uncovering frequent itemsets is usually developed by finding all the itemsets above a minimum specified threshold for the support measure defined as the frequency with which an itemset appears in the database:

$$Supp_D(X) = \frac{|t_i \in D : X \subseteq t_i|}{|D|}$$

where $X \subseteq I$ is an itemset.

In Data Streaming, the dataset varies along the time, and due to this, its size cannot be defined in advance. Thus, the problem of obtaining frequent itemsets does not only depend on the most recent transactions, but also on past transactions whose importance will depend on the time instant when they were received. Formally, the objective of FPM in stream data is to find all frequent patterns belonging to a fragment of the dataset called *time window*, noted by $W$ in the powerset of $D$. The support should be updated then along with the time, since the number of transactions in a certain window is different in each period; as a consequence, the support of an itemset will change throughout time.

Formally, a time window, $W$, is the portion of transactions of $D$ comprised between two time instants:

$$W_{i,j} = (t_i, t_{i+1}, \ldots, t_j), \quad i < j.$$

Due to the different nature of data and their heterogeneity when managing data flows, in the literature we can find different kinds of time windows. The *landmark* data model considers every data in the data stream from the beginning until the current time instant. On the other hand, the *sliding window* model considers a range of data from a fixed instant (near present) up to a certain instant in the past. The *damped window* model associates weights to the data in the stream giving higher weights to recent data than those in the past. The *time tilted window* model varies the window length and its granularity depending of its proximity to current time.

### *2.1. FPM in Static Data*

In the literature, we can find several approaches to obtain the set of frequent itemsets exceeding a predefined support threshold. Some of the most popular ones are the Apriori [10], AprioriTID [10], ECLAT [11], TreeProjection [12], and

FP-Growth approaches. There are several works comparing the performance of these algorithms [13, 14, 8]. In general, we can assume that FP-Growth is faster than the rest of algorithms.

Regarding the parallel versions, there are parallel and distributed proposals using the MapReduce paradigm. In [15, 1, 16], there are Apriori extensions for distributed environments using the Hadoop platform. The main difference of Hadoop versus using Spark is that the latter enables computations in memory, which accelerates the performance.

Among the developed proposals extending Apriori which uses the Spark platform, we find [17, 18]. These proposals are not available within the Spark libraries and the only available extension is the PFP (Parallel FP-Growth) [19]. In this case, the implementation of the tree structure employed in the FP-Growth does not scale very well within the MapReduce framework because the FP-tree may not fit in memory for very large datasets. For this reason, the parallel version available in Spark for FP-Growth is not exhaustive, because it only extracts the top-k itemsets and hence, it does not find all itemsets exceeding the support threshold.

### *2.2. FPM in Streaming Data*

In order to define the study of related works, we focused on proposals using the sliding window model, because they enable a higher flexibility to define which transactions belong to the window and facilitates the windows updating along time. Among the approaches following this model, we distinguish between sliding windows with variable and non-variable length (the amount of transactions in the window does not change).

Among the proposals using *sliding windows with non-variable length*, the CPS-Tree and the GC-Tree algorithms proposed in [20] and [21] resp. store the identified itemsets in a tree structure. The latter employs this structure to extract closed itemsets. On the contrary, in the algorithm developed in [22], the occurrences of each item are stored in a binary vector of bits to accelerate the support updating. The algorithms developed in [23, 24] extract maximal itemsets, i.e. all the supersets containing the itemset are infrequent. Despite the time saved by mining maximal itemsets, the main problem is that the entire set of frequent items cannot be recovered from the maximal itemsets.

On the other hand, the approaches using *sliding windows with variable length* are capable of adapting the window in every instant because they enable the computations with different amounts of transactions. This feature makes them more flexible towards non-period systems like transactions from social media. However, non-variable windows are appropriate for sensored systems where the periodicity of transactions remains almost constant. The algorithm proposed in [25] makes an incremental scanning to update a prefix tree structure where it stores not only the frequency of items, but also some “potential future frequent” items that are infrequent. The algorithm in [26] also uses this kind of tree structure and improves it as well by applying a greedy approach. However, it presents some drawbacks because its precision is reduced in exchange for a decreased memory consumption. This is due to the fact that items not considered potential cannot be frequent, although taking into account the transactions in the new coming windows, they would be frequent. The proposal developed in [27] also uses a prefix tree with the itemsets counts and it defines a factor index to express the relevancy of recent transactions. According to this, they make a pruning strategy for the tree which accelerates the memory consumption but decrease the computation efficiency because sometimes the counts of emerging items must be recalculated if they were discarded during the pruning phase. The approach presented in [28] introduces two tree-based data structures to add the new transactions to the original database (called a base-tree), without restructuring the base-tree for extracting frequent itemsets. However, in this work there is a lack of experimentation to know the performance of the proposed approach.

There also exist other works extracting specific types of itemsets, which is beyond the scope of this paper. Particularly, in [29] the proposed algorithm mines weighted erasable patterns from stream data proposing some prune strategies taking into account the items' weight. In [30] the work focuses on mining high utility itemsets over stream data employing for that an utility measurement.

The FIMoTS (Frequent Itemset Mining over Time-sensitive Streams) algorithm developed in [9] has several features that make it attractive for mining frequent itemsets using variable length sliding windows. We analyze it in more detail paying attention to those aspects of the algorithm that will be used for our proposal using Big Data techniques. One of the optimization techniques used is to classify the itemsets in different categories according to their support, so for every window the membership of the itemsets is revised depending on their frequency in the window. For that aim, the authors define the upper and lower bounds called *Type Transforming Upper/lower Bound*, which are different bounds associated to each itemset to represent the number of transactions that are necessary to change their classification, in particular to be frequent or infrequent. When these bounds reach zero, the support of the itemset is recalculated. Another peculiarity is that these bounds are expressed as the quotient of two integers in order to facilitate further computations. The main phases of the FIMoTS algorithm are:

- The *initialization of the enumeration tree* which contains the supports and the transforming bounds for each item. When the itemset is frequent, all its descendants are calculated taking into account the nodes at the same level and when it is not, it is kept but its descendants are not computed.

- The *aggregation and elimination of transactions to the sliding window.* When this happens, the enumeration tree must be recalculated in order to consider the new transactions of the window or the dropped transactions from it. This process consists on the computation of the transforming bounds and in the case that an itemset changes its condition from frequent to infrequent or viceversa, the descendants should be pruned or recalculated respectively.

## 3. Distributed method for frequent itemset mining in data streams

No previous revised algorithms can be directly applied in distributed clusters. Moreover, the tree structures used in all of them make their direct extension using the MapReduce framework more complicated due to the complexity and low efficiency of operating with distributed trees in different clusters.

The decision of implementing a distributed version of the FIMoTS algorithm is founded on the comparison made in [9], where this approach outperforms the rest of approaches using sliding windows. The original FIMoTS algorithm is mainly iterative and performs recurrent updates over the tree structure, but in distributive platforms, it is advisable to keep the communication among clusters as low as possible. However, the maintenance of the tree structure needs direct communication with all the clusters. For these reasons, our proposal pays much attention on this data structure and on how the exchange of information is made, making it as efficient as possible.

Taking this into account, the structures employed for our distributed approach are:

- $I$ denotes the set of items coming from the stream.
- $T_i$ represents the i-th window from the beginning of the process. The transactions collected in each window are transformed using the RDD (Resilient distributed dataset) available in Apache Spark which partitions data in order to distribute them across the clusters.
- The Frequent Itemset Tree ($FIT$), badly called tree, is a list of elements, called nodes, containing information about the frequent items found till the moment. In Figure 1 an example of node is illustrated.
- The Transforming Bounds Lists ($TBL_F$ and $TBL_I$). We define two TBLs: $TBL_F$ contains the upper and lower bounds of frequent itemsets, and $TBL_I$ the bounds for infrequent itemsets. The TBL also contains the itemsets FIT-identifiers in order to optimize their access in the FIT when the support has to be updated, i.e. when the bounds turn to 0 or less (see Figure 2).

One of the main features of FIMoTS is that it speeds up the mining process by only considering those itemsets that can change their status from frequent to infrequent and vice versa. For that aim it defines the upper and lower bounds which determines the number of transactions needed to change the itemset status:

(a) Upper bound: Given an itemset $i \in I$, if *adding* $ub - 1$ new transactions to the window it is impossible to change its status (frequent/infrequent) but adding $ub$ transactions it could change it, then $ub$ is called the *upper bound*, $ub_i$, associated to the itemset.

(b) Lower Bound: Analogously, given an itemset $i \in I$, if *removing* $lb - 1$ transactions to the window it cannot change its status (frequent/infrequent) but removing $lb$ transactions it could change it, then $lb$ is called the *lower bound*, $lb_i$, associated to the itemset.

For the computation of upper and lower bounds it is only needed the actual support of the itemset (contained in the FIT) and the number of transactions added and removed from the actual window (knowing the minimum support threshold) (see Algorithm 3). Remark that when the bounds turn to 0 or less, then the status of the itemset is changed and it has to be changed in the FIT by computing (infrequent to frequent) or removing (frequent to infrequent) its descendants.

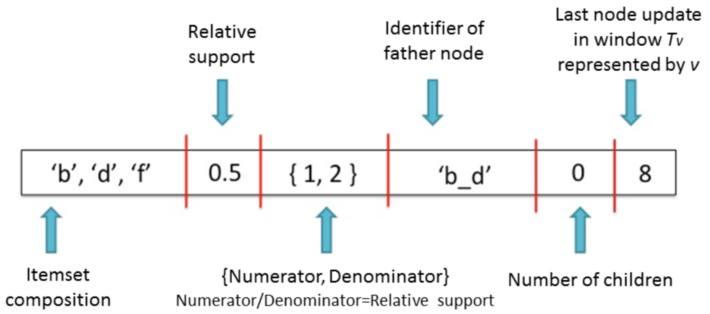

Relative support
Identifier of father node
Last node update in window $T_v$ represented by $v$
'b', 'd', 'f'
0.5
{ 1, 2 }
'b_d'
0
8
Itemset composition
{Numerator, Denominator}
Numerator/Denominator=Relative support
Number of children

Figure 1: Example of a node in the Frequent Itemset Tree

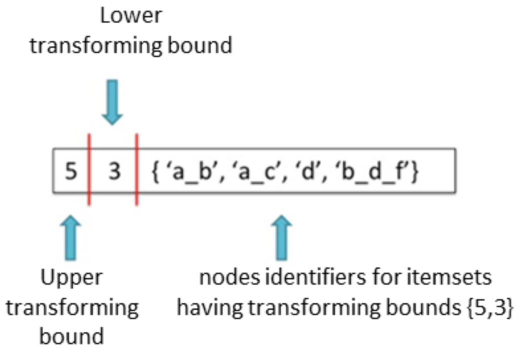

Lower transforming bound
5
3
{ 'a_b', 'a_c', 'd', 'b_d_f'}
Upper transforming bound
nodes identifiers for itemsets having transforming bounds {5,3}

Figure 2: Example of an element in the list of transforming bounds (LTB)

In Figure 3, we can see the whole picture of our proposal. In it, we distinguish the following processes:

- **Initialization of structures:** The vocabulary (set of different items) is identified and replicated in each slave machine. The transactions coming in the first window are analyzed to accomplish the tree initialization (see also Figure 4).

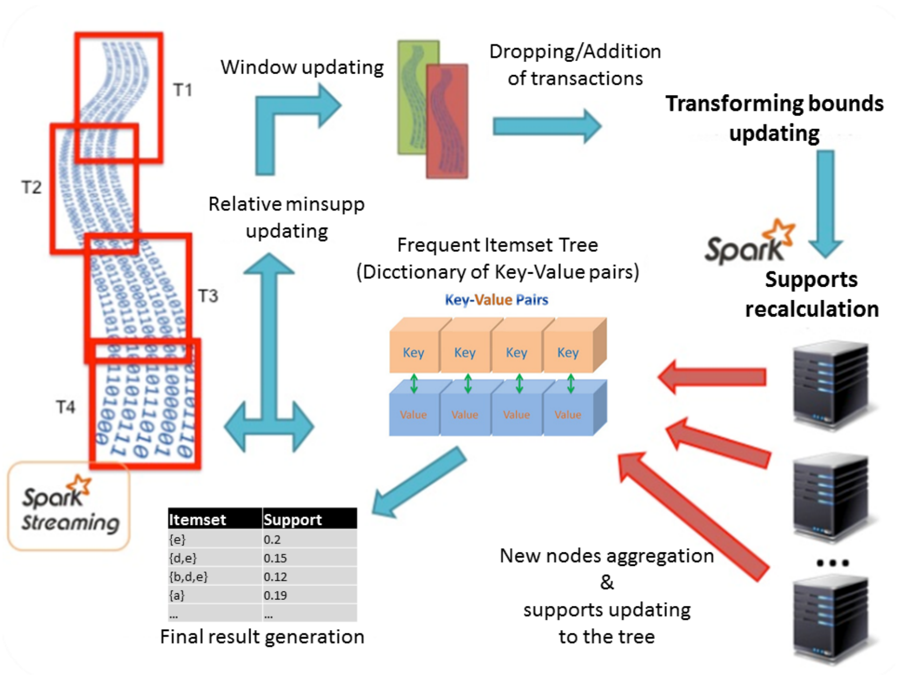
- **Data partition:** This is performed using Spark Streaming, which allows the distribution of upcoming transactions until the time interval (window) ends. Once it is closed, we can continue processing the data within it. In the meanwhile, the new coming transactions are processed and stored in an intermediate state in an RDD (Resilient Distributed Dataset) [31] which is stored in memory to be distributed and processed afterwards using the Map and Reduce functions. The main advantage of using RDDs is that we can assure a fault-tolerant system regarding fails and delays due to the internal management of RDDs in Spark [32].

- **Updates of transforming bounds:** Once the window has moved (some transactions are dropped and new coming transactions are considered), the upper and lower transforming bounds are modified without distributing the process. Since this is a numerical computation process, which does not require a big effort, it is not distributed along the clusters. Hence, it is directly computed over a list where each element is comprised of the upper transforming bound, the lower transforming bound, and all the itemsets with that pair of bounds (see Figure 2).

- **Recalculation of supports:** For those itemsets whose transforming bounds have increased/decreased to zero or low, their support is computed considering the transactions in the actual window. This computation is distributed using the MapReduce functions and is recursively applied when the descendants of the itemset must be computed, i.e. when the itemset turns from infrequent to frequent (see Algorithm 2 for a detailed description of the distributed computation of supports). In Figure 6, the process to spread out the descendants of a node that has changed to be frequent is shown.

- **Frequent itemset tree updating:** Each cluster node returns the obtained information from the distributed process in order to update the support of every itemset. This information is also used to expand or prune the frequent itemset tree. The tree structure is modeled in a global variable which is a dictionary consisting on key-value pairs, where the key is an identifier for the itemset and the value is comprised of all the information about the itemset (see Figure 1): its relative support, the quotient of transactions used for the computation of the relative support (numerator, denominator), the identifier of its father node, how many children nodes it has, and a stamp indicating its last modification. This representation of the tree enables a very rapid access from the clusters by indicating the key of the itemset, and also the removal of itemsets when the tree has to be pruned.

Figures 4 to 6 depicts some steps of the algorithm. Figure 4 corresponds to Phase 1 of Algorithm 1 where the Frequent Itemset Tree is initialized taking the information of transactions in the initial window $T_1$. First, the vocabulary is identified (lines 1-4), afterwards the transforming bounds are computed (lines


T1
T2
T3
T4
Spark Streaming
Window updating
Dropping/Addition of transactions
Transforming bounds updating
Relative minsupp updating
Frequent Itemset Tree (Dicctionary of Key-Value pairs)
Key-Value Pairs
Key
Key
Key
Key
Value
Value
Value
Value
Spark
Supports recalculation
New nodes aggregation & supports updating to the tree
Final result generation

| Itemset | Support |
|---|---|
| {e} | 0.2 |
| {d,e} | 0.15 |
| {b,d,e} | 0.12 |
| {a} | 0.19 |
| ... | ... |

Figure 3: Overall scheme of our proposal

5-7) using the procedure of Algorithm 3. For those items in $LTB_F$ (see Figure 2) it is first computed their support and stored as a node in the Frequent Itemset Tree (lines 8-11). Then, the FIT is spread out with the rest of descendant nodes following procedure described in lines 12-22. This part is depicted in Figure 6. In Phase 2 of Algorithm 1 is executed when a new window arrives with new transactions. It encompasses mainly 3 steps: $MinSupp$ recomputation, with that we can update the list of transforming bounds with the information of transactions in the new window, and finally we compute the itemsets support and update the Frequent Itemset Tree. Phase 3 is executed when we want to obtain the frequent itemsets found till the moment by checking the Frequent Itemset Tree. Algorithm 2 details how it is computed the itemsets support using Spark (see also Figure 5).

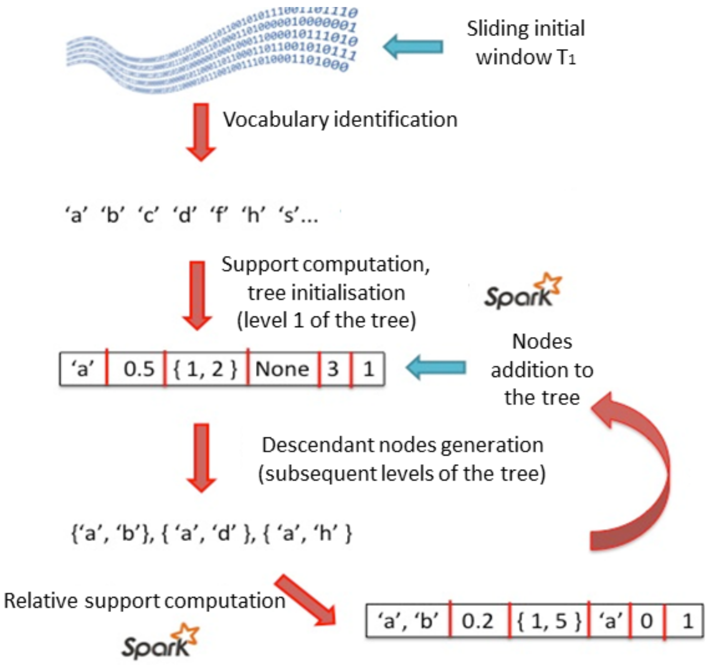

Sliding initial window T1
Vocabulary identification
'a' 'b' 'c' 'd' 'f' 'h' 's'...
Support computation, tree initialisation (level 1 of the tree)
Spark
Nodes addition to the tree
'a' | 0.5 | { 1, 2 } | None | 3 | 1
Descendant nodes generation (subsequent levels of the tree)
{'a', 'b'}, { 'a', 'd' }, { 'a', 'h' }
Relative support computation
Spark
'a', 'b' | 0.2 | { 1, 5 } | 'a' | 0 | 1

Figure 4: Overall scheme of the Frequent Itemset Tree initialization

## 4. Experimental evaluation

The experiments carried out have been designed to analyze two different aspects:

- To compare our proposal with the non-distributive version of FIMoTS taking into account the execution time and the memory consumption.

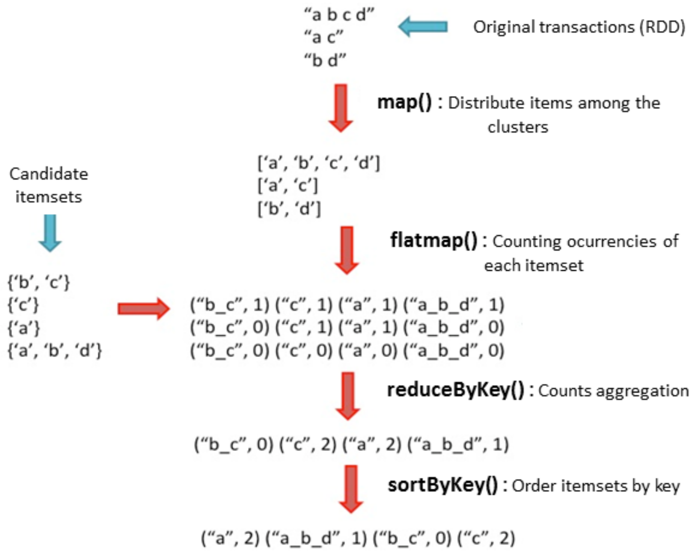

"a b c d"
"a c"
"b d"
Original transactions (RDD)
map() : Distribute items among the clusters
['a', 'b', 'c', 'd']
['a', 'c']
['b', 'd']
Candidate itemsets
{'b', 'c'}
{'c'}
{'a'}
{'a', 'b', 'd'}
flatmap() : Counting ocurrencies of each itemset
("b_c", 1) ("c", 1) ("a", 1) ("a_b_d", 1)
("b_c", 0) ("c", 1) ("a", 1) ("a_b_d", 0)
("b_c", 0) ("c", 0) ("a", 0) ("a_b_d", 0)
reduceByKey() : Counts aggregation
("b_c", 0) ("c", 2) ("a", 2) ("a_b_d", 1)
sortByKey() : Order itemsets by key
("a", 2) ("a_b_d", 1) ("b_c", 0) ("c", 2)

Figure 5: Distributed computation of itemsets support using Spark

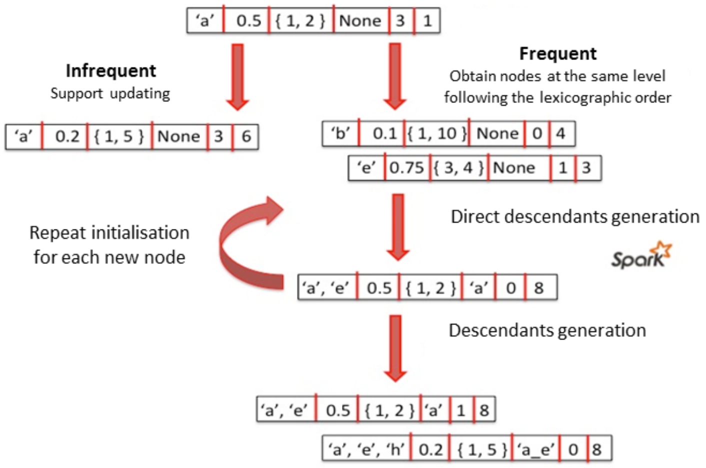

'a' | 0.5 | { 1, 2 } | None | 3 | 1
Infrequent
Support updating
'a' | 0.2 | { 1, 5 } | None | 3 | 6
Frequent
Obtain nodes at the same level following the lexicographic order
'b' | 0.1 | { 1, 10 } | None | 0 | 4
'e' | 0.75 | { 3, 4 } | None | 1 | 3
Direct descendants generation
Repeat initialisation for each new node
Spark
'a', 'e' | 0.5 | { 1, 2 } | 'a' | 0 | 8
Descendants generation
'a', 'e' | 0.5 | { 1, 2 } | 'a' | 1 | 8
'a', 'e', 'h' | 0.2 | { 1, 5 } | 'a_e' | 0 | 8

Figure 6: Spread out of Frequent Itemset Tree by items descendants generation

**Algorithm 1** : Main procedure for Distributed FIMoTS

```
    Input: Data: an RDD of transactions.
    Input: $MinSupp$: Minimum support threshold as a fraction $a/|T_i|$.
    Output: Frequent itemsets exceeding MinSupp
    Phase 1: Initialization of structures
 1: for all $t \in T_1$ do        // vocabulary identification
 2:     if $i \in t$ and $i \notin I$ then $I \leftarrow i$
 3:     end if
 4: end for
    LTBs and FIT initialization
 5: for all $item \in I$ do
 6:     $LTB_F, LTB_I \leftarrow$ LTB computation($item, 0, |T_1|, MinSupp$)
 7: end for
 8: for all $item \in LTB_F$ do
 9:     $Result \leftarrow$ Distributed Support computation ($T_1$,$\{i \in I : i \in LTB_F\}$)
        // FIT node computation
10:     FIT $\leftarrow < item, frequency/|T_1|, \{frequency, |T_1|\}, 0, 0, 1 >$
        //no father node, no children nodes, Identifier of Window=1
11: end for
    Descendant FIT nodes generation
12: while $LTB_F$ has itemsets do
13:     for all $item \in I : item \geq MinSupp$ do
            //Candidates generation
14:         $I \leftarrow C_k = item \cup itemset$ s.t. $item \cap itemset = \emptyset$
15:         $LTB_F, LTB_I \leftarrow$ Compute LTBs for the new candidate (Step 6)
16:     end for
17: end while
    //Optimization
18: Result $\leftarrow$ Distributed Support computation ($T_i$,$\{C_k \in LTB_F\}$)
19: for all itemset $\in$ Result do        // FIT node descendants computation
20:         FIT $\leftarrow < itemset, frequency/|T_1|, \{frequency, |T_1|\}, f, 0, 1 >$
            // updating father nodes: adding its children
21:         FIT $\leftarrow < itemset, frequency/|T_1|, \{frequency, |T_1|\}, f, c, 1 >$
22: end for
    Phase 2: Structures Updating
23: $MinSupp$ recalculation (as a fraction) using new and removed transactions
    in $T_k$
24: LTB Updating for all itemsets (similar to step 9 or 15)
    //Supports computation of those itemsets changing from frequent to infrequent and vice versa
25: if $LTB_F(itemset).upperBound \leq 0$ or $LTB_F(itemset).lowerBound \leq 0$
    then
26:     Result $\leftarrow$ Distributed Support computation ($T_k$,$\{i \in LTB_F\}$)
        //FIT Updating and Spread out
27:     Repeat steps 19-22 to update FIT taking into account Result
28: end if
    Phase 3: Results visualization
29: Frequent_itemsets $\leftarrow$ FIT
```

**Algorithm 2** : Distributed Support Computation

```
Input: $T_i$ : Window containing the transactions in RDD.
Input: $I$: Candidate itemsets.
Output: Pairs of items/itemsets with their frequency in $T_i$

$data\_map \leftarrow$ map( $t \in T_i$)
        $l \leftarrow Split(t)$
    end map
$data\_map \leftarrow$ flatmap $l \in data\_map$
        $l \leftarrow getItemsets(l)$
        //Returns a list with pairs of the form $< itemset, 1 >$
    end flatmap
$Result \leftarrow$ reduce $l \in data\_map$
        $Result \leftarrow < itemset, frequency > \leftarrow ReduceByKey(itemset)$
    end reduce
$Result \leftarrow SortBykey(Result)$
```

**Algorithm 3** : LTB computation

```
Input: $itemset$.
Input: $rt$ : Number of transactions removed from the Window.
Input: $at$: Number of transactions added to the Window.
Input: $MinSupp$: Minimum support threshold as a fraction $(a/|T_i|)$.
Output: Updated Lists of Transforming Bounds

if itemset is frequent then
    $LTB_F(itemset).upperBound \leftarrow itemset.upperBound - at - \frac{|T_i|-a}{a}rt$
    $LTB_F(itemset).lowerBound \leftarrow itemset.lowerBound - rt - \frac{a}{|T_i|-a}at$
else        // itemset infrequent
    $LTB_I(itemset).upperBound \leftarrow itemset.upperBound - at - \frac{a}{|T_i|-a}rt$
    $LTB_I(itemset).lowerBound \leftarrow itemset.lowerBound - rt - \frac{|T_i|-a}{a}at$
end if
```

- To study the behavior of our proposal towards different configurations and datasets.

The experiments performed are aimed not only to compare a sequential vs a distributive version, which in theory should be better, but also to highlight that using a distributed version outperforms existent approaches for mining frequent itemsets in stream data (without needing the data storage during the process since only the Frequent Itemset Tree is preserved). This is very useful in many real applications where the quantity of data to be analyzed is huge and is being continuously generated (e.g. social media, economy, sensors, etc.).

As far as we know, there are no available Big Data implementations for frequent itemset stream mining. Additionally, it is also worth to stress that in [9] the reader can find an extensive comparison among the non-distributed algorithms for stream frequent itemset mining, obtaining that FIMoTS outperforms the rest of approaches using sliding windows. For this reason, we compare our proposal with the sequential version of FIMoTS, trying to solve the limitations (memory and time) that the FIMoTS presents when massive streaming data sets are considered.

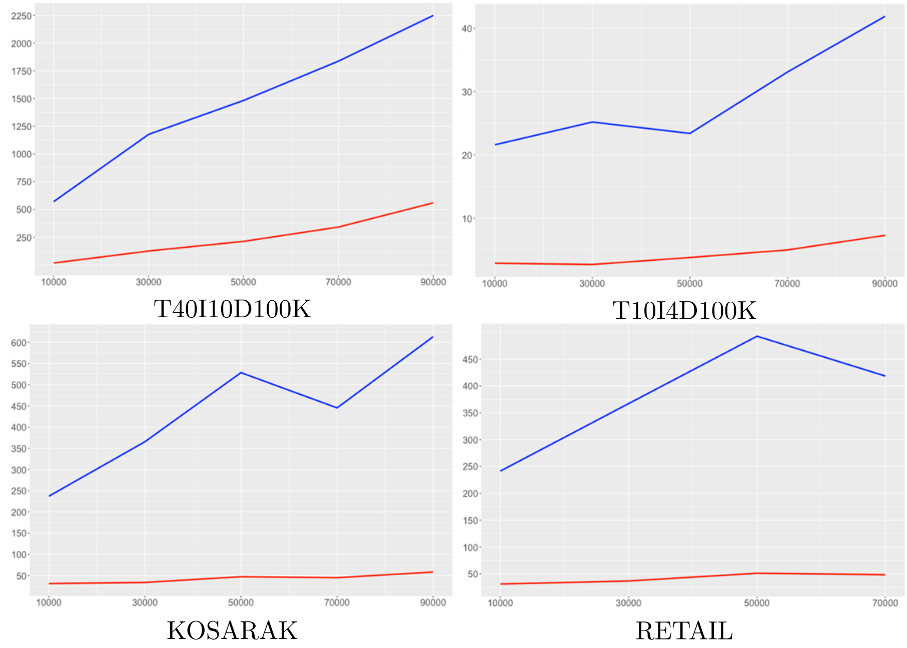
Among the chosen datasets we have selected those used in [9] in order to recreate the results of FIMoTS. These datasets present different features: wide/limited number of transactions and itemsets, low/high correlation of transactions with respect to the vocabulary (items), different length of transactions, synthetic/real data, etc. Datasets T10I4D100K and T40I10D100K have been artificially generated by the IBM tool, KOSARAK comprises data about the clicks made in a news site from Hungary, and RETAIL contains information about the purchased products in a Belgian outlet.

We have conducted all the experiments on a 64-bits architecture with 32 cores (although only 16 of them were used) on 2 Intel Xeon E5 and with 30 GB of RAM functioning over an operative system with CentOS 6.8. The Spark version was 1.5 using a fully distributed mode with Cloudera Manager.

Table 1 shows the details about the datasets used in the experiments. The second and third columns respectively contain the total number of transactions and items analyzed. The following three columns represent the average length of items in the transaction, the minimum number of items, and the maximum number of items in a transaction, respectively. The last column represents the proportion computed as the quotient $v/l$, being $v$ the total number of items (vocabulary) and $l$, the average length of items in a transaction.

| Dataset | Total trans. | Total items | Average | Min | Max | Proportion (v/l) |
|---|---|---|---|---|---|---|
| KOSARAK | 990002 | 36 841 | 7.1 | 1 | 2497 | 5188.8 |
| T10I4D100K | 100000 | 870 | 10.1 | 1 | 29 | 86.1 |
| T40I10D100K | 100000 | 942 | 39.6 | 4 | 77 | 23.8 |
| RETAIL | 88162 | 16470 | 10.3 | 1 | 76 | 1598.1 |

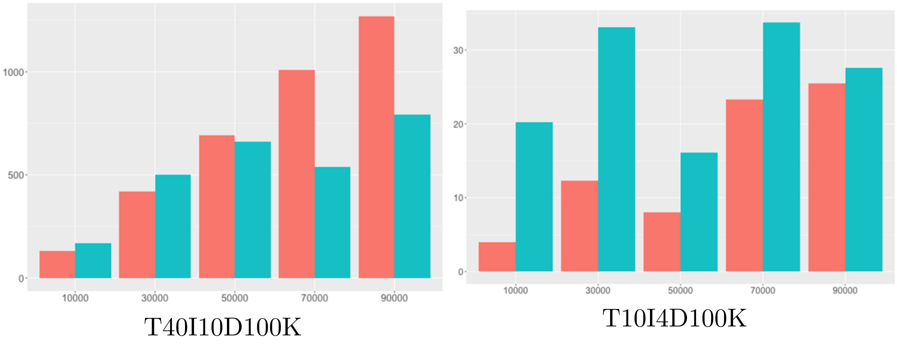
Table 1: Dataset description



T40I10D100K T10I4D100K

KOSARAK RETAIL

Figure 7: Average time in seconds (y-axis) for different window lengths (x-axis). Red line represents our proposal and blue line, the non-distributed FIMoTS



T40I10D100K T10I4D100K

Figure 8: Memory consumed in MB (y-axis) for different window lengths (x-axis). Red represents our proposal and blue, the non-distributed FIMoTS

Figure 7 shows the time spent by original FIMoTS and our distributive proposal for the different datasets using different window lengths (x-axis). It is worth mentioning that for the KOSARAK and the RETAIL datasets the experimentation could not be finalized, in the non-distributed case, when larger windows (number of transactions) were considered due to memory overflow in both cases. This is mainly due to the large number of items in these datasets which cannot be handled at the same time in main memory. It is important to note that, in all cases, the distributed version outperformed the original FIMoTS, as expected.

Regarding memory consumption, Figure 8 shows the resulting memory usage for the synthetic datasets (since for KOSARAK and RETAIL, some experiments were not concluded). In the left graph, the behavior of our approach is worse when increasing the window lengths. This is due to the high quantity of infrequent itemsets that have to be maintained in the FIT in case they turn to be frequent in a posterior moment. On the contrary, in the right graph, we can see that the memory consumed by the distributed version improves the memory consumption in all cases, although the improvement is lower and it decreases as the window size increases.

Concerning the performance of our proposal, we have carried out several experiments for different configurations of the *minsupp* threshold and analyzed the time taken and the memory consumed in the initialization phase and for the updating of the results in average for all the windows. For being able to compare the results, we have fixed the window length to 10000 transactions. In Tables 2 -5, we can see the results obtained in each of the datasets. From these results, we can see that for the initialization tree process, the spent time is higher than its updating when a new window is considered. This is due to the fact that the algorithm must compute all the transforming bounds and supports of itemsets, while during the updating phase only part of the tree is recalculated. Moreover, the execution time spent during the updating process remains more or less constant in each incoming package even when the amount of itemsets is very high (see the cases for the minimum support of 0.01). Finally, the memory employed in each case remains proportional to the quantity of itemsets processed (last column in the tables).

| *minsupp* | initialization /updating packages | frequent/ infrequent itemsets | initialization/ average updating time | initialization/ average updating memory |
|---|---|---|---|---|
| 0.01 | 10 / 90 | 780 / 126098 | 931 / 853.66 s | 25.39 MB/ 2.53 GB |
| 0.05 | 10 / 90 | 20 / 938 | 22 / 6 s | 128.4 KB/12.84 MB |
| 0.1 | 10 / 90 | 1 / 866 | 12 / 2.7 s | 40.11 KB/ 4.01 MB |

Table 2: Some analytics for several *misupp* thresholds with 10000 transactions windows length in T10I4D100K

| *minsupp* | initialization /updating packages | frequent/ infrequent itemsets | initialization/ average updating time | initialization/ average updating memory |
|---|---|---|---|---|
| 0.01 | 10 / 90 | 1654/135467 | 1576/1254 s | 28.76 MB/ 2.87 GB |
| 0.05 | 10 / 90 | 628 / 89906 | 687 / 804.9 s | 17.08 MB/ 1.7 GB |
| 0.1 | 10 / 90 | 160 / 6987 | 118 / 74.7 s | 1.31 MB/131.4 MB |

Table 3: Some analytics for several *misupp* thresholds with 10000 transactions windows length in T40I10D100K

| *minsupp* | initialization /updating packages | frequent/ infrequent itemsets | initialization/ average updating time | initialization/ average updating memory |
|---|---|---|---|---|
| 0.01 | 10 / 981 | 127 / 13585 | 78 / 65.4 s | 361.3 KB/361.3 MB |
| 0.05 | 10 / 981 | 33 / 10128 | 41/ 38.1 s | 283.88 KB/283.88 MB |
| 0.1 | 10 / 981 | 9 / 10092 | 40/ 30.8 s | 254.37 KB/254.37 MB |

Table 4: Some analytics for several *misupp* thresholds with 10000 transactions windows length in KOSARAK

| *minsupp* | initialization /updating packages | frequent/ infrequent itemsets | initialization/ average updating time | initialization/ average updating memory |
|---|---|---|---|---|
| 0.01 | 10 / 79 | 443 / 18261 | 92 / 253 s | 8.96 MB/797.44 MB |
| 0.05 | 10 / 79 | 37 / 8609 | 30/ 33.3 s | 4.04 MB/359.56 MB |
| 0.1 | 10 / 79 | 21 / 8612 | 31/ 30.8 s | 3.97 MB/354.09 MB |

Table 5: Some analytics for several *misupp* thresholds with 10000 transactions windows length in RETAIL

## 5. Conclusions

The developed proposal is intended to enable the tendency analysis in massive amounts of data coming from data streams. This is extremely useful in areas where the amount of generated information exceeds the usual bounds of static databases and is in continuous movement. These are the cases of social media (Twitter, Linkedin, Instagram, etc.) or economical/business analysis.

In this work, we focus in frequent itemset mining for finding the most relevant tendencies in data taking into account their appearance. We have revised the existent algorithms in frequent itemset mining taking into account both perspectives: massive data and data coming from streams. The best to our knowledge, there is no approach for frequent itemset mining taking into account both premises.

We have therefore developed an algorithm capable of extracting frequent itemsets in continuous flows of data by using the MapReduce framework and the Spark Streaming platform. The experiments carried out show that, as it was expected, our proposal outperforms the non-distributed version of the algorithm, and moreover, some experiments which could not be finished by the original

FIMoTS can now be executed. It is worth to note that our approach can be executed over variable windows length and support thresholds that change during the algorithm execution without initializing the Frequent Itemset Tree nor the Bounds lists LTBs. This is extremely useful in real streaming data where it is not known apriori the quantity of new data coming in each moment.

In future works, we plan to apply the developed algorithm for social media analysis in real time and extend it to consider association rule mining in order to study the co-occurrences of frequent items in data streams.

## Acknowledgement

The research reported in this paper was partially supported by the Spanish Ministry for Economy and Competitiveness by the project grant TIN2015-64776-C3-1-R, the Andalusian Government (Junta de Andalucía) under project P11-TIC-7460 and the COPKIT project from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 786687).

## References

[1] Z. Farzanyar, N. Cercone, Efficient mining of frequent itemsets in social network data based on mapreduce framework, in: Proceedings of the 2013 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013), 2013, pp. 1183–1188.

[2] J. Han, E. Haihong, G. Le, J. Du, Survey on NoSQL database, in: Pervasive computing and applications (ICPCA), 2011 6th international conference on, IEEE, 2011, pp. 363–366.

[3] D. Borthakur, HDFS architecture guide, HADOOP APACHE PROJECT http://hadoop. apache. org/common/docs/current/hdfs design. pdf.

[4] J. Dean, S. Ghemawat, MapReduce: simplified data processing on large clusters, Communications of the ACM 51 (1) (2008) 107–113.

[5] T. White, Hadoop: The definitive guide, " O'Reilly Media, Inc.", 2012.

[6] H. Karau, A. Konwinski, P. Wendell, M. Zaharia, Learning Spark: Lightning-Fast Big Data Analysis, " O'Reilly Media, Inc.", 2015.

[7] S. Singh, R. Garg, P. Mishra, Performance analysis of apriori algorithm with different data structures on hadoop cluster, International Journal of Computer Applications 128 (9) (2015) 45–51.

[8] K. Garg, D. Kumar, Comparing the performance of frequent pattern mining algorithms, International Journal of Computer Applications 69 (25).

[9] H. Li, N. Zhang, J. Zhu, H. Cao, Y. Wang, Efficient frequent itemset mining methods over time-sensitive streams, Knowledge-Based Systems 56 (2014) 281 298.

[10] R. Agrawal, R. Srikant, et al., Fast algorithms for mining association rules, in: Proc. 20th int. conf. very large data bases, VLDB, Vol. 1215, 1994, pp. 487–499.

[11] M. J. Zaki, S. Parthasarathy, M. Ogihara, W. Li, et al., New algorithms for fast discovery of association rules., in: KDD, Vol. 97, 1997, pp. 283–286.

[12] R. Agarwal, C. C. Aggarwal, V.V.V.Prasad, A tree projection algorithm for generation of frequent itemsets, Journal of Parallel and Distributed Computing 61 (2001) 350371.

[13] J. Hipp, U. Güntzer, G. Nakhaeizadeh, Algorithms for association rule mining - a general survey and comparison, ACM sigkdd explorations newsletter 2 (1) (2000) 58–64.

[14] D. Hunyadi, Performance comparison of Apriori and FP-Growth algorithms in generating association rules, in: Proceedings of the European computing conference, 2011, pp. 376–381.

[15] N. Li, L. Zeng, Q. He, Z. Shi, Parallel implementation of apriori algorithm based on mapreduce, in: Proceedings of the 2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD '12, IEEE Computer Society, Washington, DC, USA, 2012, pp. 236–241.

[16] Z. Farzanyar, N. Cercone, Accelerating frequent itemset mining on the cloud: A mapreduce-based approach, in: IEEE 13th International Conference on Data Mining Workshops, 2013, pp. 592–598.

[17] H. Qiu, R. Gu, C. Yuan, Y. Huang, Yafim: A parallel frequent itemset mining algorithm with spark, in: Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International, IEEE, 2014, pp. 1664–1671.

[18] S. Rathee, M. Kaul, A. Kashyap, R-apriori: An efficient apriori based algorithm on spark, in: Proceedings of the PIKM'15, ACM, Melbourne, VIC, Australia, 2015.

[19] H. Li, Y. Wang, D. Zhang, M. Zhang, E. Y. Chang, PFP: parallel fp-growth for query recommendation, in: Proceedings of the 2008 ACM conference on Recommender systems, ACM, 2008, pp. 107–114.

[20] S. Tanbeer, C. Ahmed, B. Jeong, Y. Lee, Sliding window-based frequent pattern mining over data streams, Information sciences 179 (22) (2009) 3843–3865.

[21] J. Chen, S. Li., GC-tree: a fast online algorithm for mining frequent closed itemsets, 2007, pp. 457–468.

[22] H. Li, C. Ho, S. Lee, Incremental updates of closed frequent itemsets over continuous data streams, Expert Systems with Applications 36 (2) (2009) 2451–2458.

[23] H. Li, N. Zhang, A false negative maximal frequent itemset mining algorithm over stream, 2011, pp. 29–41.

[24] H. Li, N. Zhang, Z. Chen, A simple but effective maximal frequent itemset mining algorithm over streams, Journal of Software 7 (1) (2012) 25–32.

[25] H. Li, S. Lee, Mining frequent itemsets over data streams using efficient window sliding techniques, Expert Systems with Applications 36 (2) (2009) 1466–1477.

[26] J. Koh, Y. Don, Approximately mining recently representative patterns on data streams, in: Proceedings of PAKDD Pacific-Asia Conference on Knowledge Discovery and Data Mining, Vol. 4819, 2007, pp. 231–243.

[27] H. Chen, L. Shu, J. Xia, Q. Deng, Mining frequent pattern in varying-size sliding window of online transactional data streams, Information Sciences 215 (2012) 15–36.

[28] P. Pimpale, N. Rakhecha, M. Saindane, H. Sawant, Real-time stream data mining to find frequent item-set, International Journal of Computer Science and Mobile Computing, IJCSMC 6 (4) (2017) 134–137.

[29] U. Yun, G. Lee, Sliding window based weighted erasable stream pattern mining for stream data applications, Future Generation Computer Systems 59 (2016) 1–20.

[30] H. Ryang, U. Yun, High utility pattern mining over data streams with sliding window technique, Expert Systems with Applications 57 (15) (2016) 214–231.

[31] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. Franklin, S. Shenker, I. Stoica, Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing, in: 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2012.

[32] M. Zaharia, T. Das, H. Li, T. Hunter, Discretized streams: Fault-tolerant streaming computation at scale, in: ACM Symposium on Operating Systems Principles (SOSP13), 2013, pp. 423–438.

# 4 Visualization tool for association rules

## 4.1 A visualization methodology for association rules through an intermediate representation (Submitted to Knowledge-Based Systems)

- Carlos Fernandez Basso, M. Dolores Ruiz, Maria J. Martin-Bautista.
  - Status: **Submitted to Knowledge-Based Systems**.
  - Impact Factor (JCR 2018): **5.101**
  - Subject Category: **Computer Science, Information Systems**
  - Rank: **27/155**
  - Quartile: **Q1**

# An intermediate representation to support association rules visualization

Carlos Fernandez-Basso[a,*], M. Dolores Ruiz[b], Miguel Molina-Solana[c,a], Maria J. Martin-Bautista[a]

[a] *Dept. Computer Science and AI, CITIC-UGR, University of Granada, Spain*
[b] *Dept. of Statistics and Operations Research, University of Granada, Spain*
[c] *Data Science Institute, Imperial College London, UK*

**Abstract**

Data mining techniques are nowadays highly useful and widely used in industry, business and government. However, their broad adoption is sometimes limited because non-expert users are not able to rightly interpret and deal with the complex results obtained. In this paper, we put forward a methodology for the display of association rules using an intermediate form. This technique enables an efficient processing of the rules by generating a standard format through a graph structure that allows us to adapt the rules to different display tools. We also show some illustrative examples of the usefulness of this intermediate form.

*Keywords:* Association Rules, Visualization, Frequent itemsets mining, Data Mining

## 1. Introduction

Many large companies, sensorized buildings and social networks generate large amounts of data on a daily basis. All these types of data (tweets about a particular topic, temperature measures in different rooms, or sales in an online shop) can be analyzed using Data Mining techniques. Among them, the extraction of association rules [1] enables the identification of relevant patterns (and thus, useful information) in a non-supervised way, from huge data sets such as sensors [2], recommendation systems [3], social networks [4] or purchases and events in web systems [5].

Within these data science applications, frequent itemset and association rule mining are specially interesting to search for relationships between events, courses or opinions [6]. On one hand, frequent itemset algorithms can be used

[*] Corresponding author

*Email addresses:* cjferba@decsai.ugr.es (Carlos Fernandez-Basso), mariloruiz@ugr.es (M. Dolores Ruiz), mmolinas@ic.ac.uk (Miguel Molina-Solana), mbautis@decsai.ugr.es (Maria J. Martin-Bautista)

to discover other types of patterns such as sequential patterns [7], gradual dependencies [8] or exception and anomalous rules [9] to discover meaningful and different types of relationships among them. On the other hand, association rule algorithms allow us to discover relationships between items in the database. In both cases, the understandability and interpretation of the extracted information for experts or end-users is crucial to maximize the usefulness of the knowledge extracted in the set of discovered rules, especially when we have large data sets, since these algorithms may generate results with a huge number of rules. Moreover, the storage of this type of results is often inefficient and in most cases does not allow its use by different display tools because there is no standard format. To solve these problems, visualization techniques are tremendously helpful to observe the relationships between the items through the rules in a simple and interpretative way.

In the literature, we can find several display applications [10, 11, 12], in particularly for the rules of association [13, 14]. However, all of them have a the lack of interoperability due to a missing formal representation of association rules. For these reasons, in this paper we present a methodology for visualizing association rules using an intermediate form to improve the interoperability. Moreover, with this methodology we also enable a faster processing of rules that better adapts to our necessities before choosing the visualization technique.

This paper is structured as follows: Section 2 explains the basic concepts about association rules and frequent itemsets. In Section 3, we study the different tools in the literature to visualize association rules. To do this, we analyze the workflow used to extract and visualize association rules. In Section 4, we propose a new methodology to visualize association rules through an intermediate representation, and provide some examples of visualization employing the intermediate representation. Finally, we present some conclusions and future works.

## 2. Preliminary concepts

This section introduces all the major foundational concepts required to understand the notions related to association rules presented in this paper.

### *2.1. Association rules*

Association rules were formally defined for the first time by Agrawal et al. [15]. Let $D = t_1, t_2, t_3..., t_n$ be a transactional database where each transaction is a subset of items from $I = i_1, ..., i_m$. Then, the problem consists in discovering implications of the form $X \rightarrow Y$ where $X, Y$ are subsets of items from $I = i_1, i_2, ..., i_m$ fulfilling that $X \cap Y = \emptyset$, where each transaction contains subsets of items from $I$. $X$ is usually referred as the antecedent (or left hand side of the rule) and $Y$ as the consequent (or right hand side) of the rule.

The most commonly used measures to extract frequent itemsets and association rules are:

- The *support* [16] which is the measure of the frequency with which an item appears in the database. In general, the most interesting association rules are those with a high support value. The support of an item is defined as follow:
$$Supp_D(X) = \frac{|t_i \in D : X \subseteq t_i|}{|D|} \tag{1}$$
The support of a rule $X \rightarrow Y$ is computed as $Supp_D(X \cup Y)$.

- The *confidence*. Given the itemsets $X$ and $Y$, and the database $D$, the confidence of an association rule $X \rightarrow Y$ [16], represented as $Conf_D(X \rightarrow Y)$, is the conditional probability of $Y$ appearing in those transactions in $D$ that contain $X$.
$$Conf_D(X \rightarrow Y) = \frac{Supp_D(X \cup Y)}{Supp_D(X)} \tag{2}$$

The problem of uncovering association rules is usually developed in two steps:

- Step 1: Finding all the itemsets above the minimum support threshold. These itemsets are known as frequent itemsets.

- Step 2: Using these frequent itemsets, association rules are discovered by imposing a minimum threshold for an assessment measure such as the confidence measure.

## 3. Association rules visualization. Previous works and comparison

This section is devoted to present basic concepts and terminology from visualization and data structure models, and review previous works and applications for visualizing association rules. We also explain the principal advantages and drawbacks of these tools and analyze their interoperability and interpretability. To this end, the different tools found in the literature have been classified in two different ways: 1) according to the type of visualization they enable to perform, and 2) depending on their functionalities (i.e. if the tool enables only visualization or it also includes the frequent itemset or association rule discovery process).

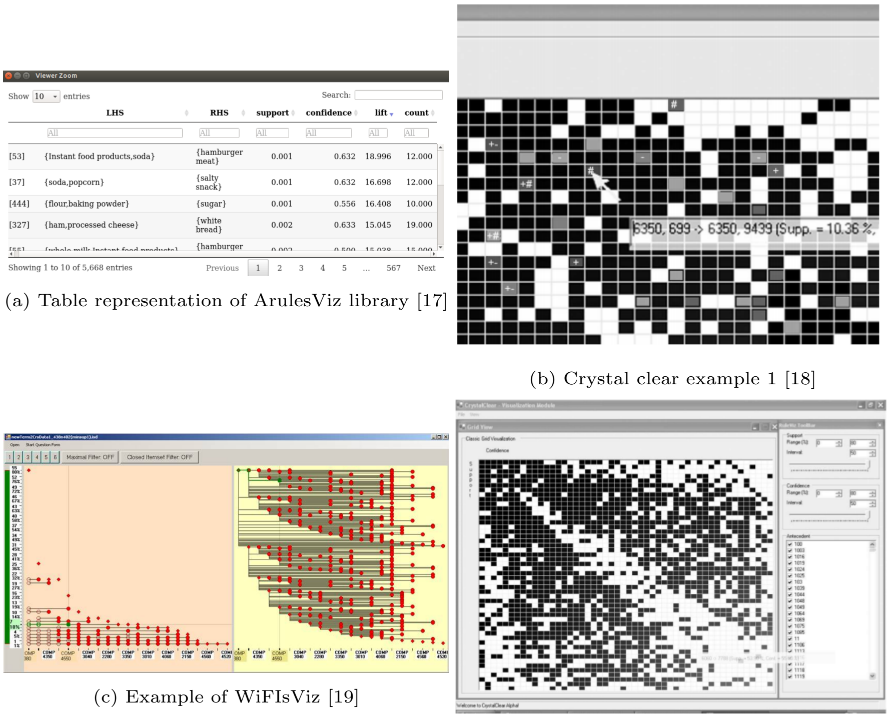
### *3.1. Classification by type of display*

Visualization of association rules can be used in different ways, as we have seen above, depending on the capabilities implemented by the tool, or by the type of visualization they implement. However, we must also take into account the structure of the results obtained and what the user wants to observe in the graphical representation in order to facilitate the results interpretation and to discover hidden information. These techniques can be classified in different subgroups according to the different ways and structures used to represent association rules. Consequently, we analyzed the tools for visualizing association rules attending to different categories, such as the data structure or the kind of visualization. In the following sections, we divide the available visualization tools into several groups: tabular representation, parallel coordinates, visualization using matrices, trees, and graph-based methods.

### *3.1.1. Tabular and grid visualization*

One method for visualizing association rules is to use a tabular representation. Although it is a simple representation, it is useful for small sets of rules that may have been selected using other display or search methods. In addition, this type of method allows the end-user to observe the rule sets sorted according to different criteria, e.g. taking into account their assessment measures, such as support, confidence or lift for example.

Different proposals for tabular representation can be seen in Figure 1. In particular, in Figure 1a we can find a table representation through the *ArulesViz* library [17]. With this representation, it is easy to see the rules, and their measures and is very simple to rank them according to the value of these measures, e.g. support, confidence, etc. However, if the number of rules is large or if we want to explore the relationship of items or itemsets, the tabular representation is not ideal.



Show 10 entries Search:

| | LHS | RHS | support | confidence | lift | count |
|---|---|---|---|---|---|---|
| | All | All | All | All | All | All |
| [53] | {Instant food products,soda} | {hamburger meat} | 0.001 | 0.632 | 18.996 | 12.000 |
| [37] | {soda,popcorn} | {salty snack} | 0.001 | 0.632 | 16.698 | 12.000 |
| [444] | {flour,baking powder} | {sugar} | 0.001 | 0.556 | 16.408 | 10.000 |
| [327] | {ham,processed cheese} | {white bread} | 0.002 | 0.633 | 15.045 | 19.000 |
| [55] | {whole milk,Instant food products} | {hamburger | 0.002 | 0.500 | 15.038 | 15.000 |

Showing 1 to 10 of 5,668 entries Previous 1 2 3 4 5 … 567 Next

(a) Table representation of ArulesViz library [17]

(b) Crystal clear example 1 [18]

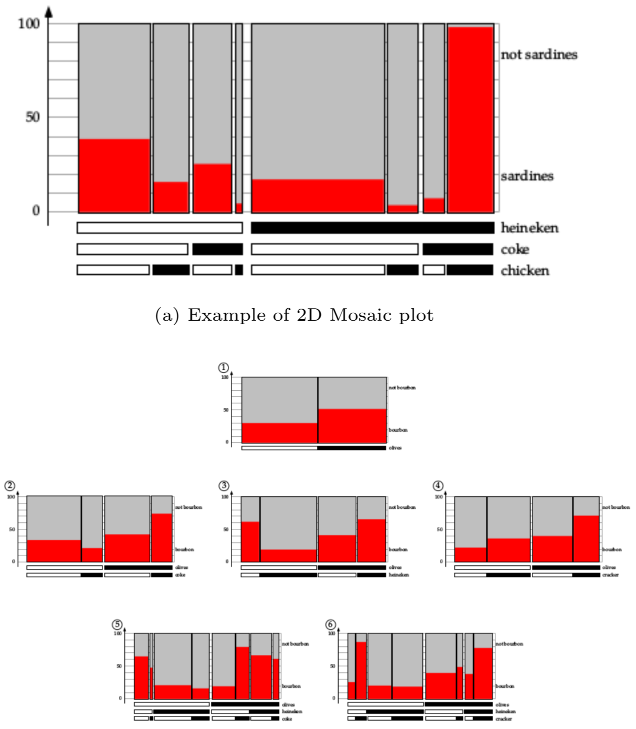
(c) Example of WiFIsViz [19]

(d) Crystal clear example 2 [18]

Figure 1: Tabular-based examples

Another tabular way of representation is by means of a grid where we can see the rules represented textually or using some edges to connect them. It is possible to observe these two different methods in Figures 1b, 1c, 1d. The first of them (1b) is called *Cristal clear* [18], which allows to represent the rules in an orderly manner according to their measures. However, its main drawback is that you need to position yourself in the cell to discover the rule and its measures, a process that makes it quite difficult and complex to explore the rules. Additionally, if the number of rules is large we may find that the cells are too small.

In the second option (Figure 1c), the tool used is WiFIsViz [19]. In this method, the rules are represented by some kind of tabular tree where items are placed in the axes. In this case the relationship between consequent and antecedent is more intuitive. Besides, the thickness of the nodes represents the measured strength of the selected rule (trust, support, lift...). This characteristic allows a better observation of higher value measures easily. In this case the main drawback is that when the number of rules increases the representation becomes very difficult to interpret and understand because the trees overlap in some nodes and edges.



(a) Example of 2D Mosaic plot

(b) Examples of mosaic plots for a set of rules

Figure 2: Mosaic plots examples [20]

Another proposal can be found in [20], where rules are represented in the form of mosaic plots. In this case, the way of visualizaing the rules and their measure of interest is transformed through contingency tables. It can be observed in Figure 2, that this visualization method displays items in a clearer and more intuitive way than the previous tabular-based visualization methods. Although its way of visualization works well for rules with few items, for many elements it quickly loses interpretability.

The main advantages of these types of tabular display techniques are their simplicity, ease of use and the elimination of new evaluation measures. This allows us to classify the results by their evaluation values. Because tables are a very common representation, they are generally very intuitive for any end user who wants to explore the results. Therefore, it is a convenient complementary option. They are also a good solution if we have the possibility to filter, order and select rules from the results and use this kind of representations to explore them in more detail.

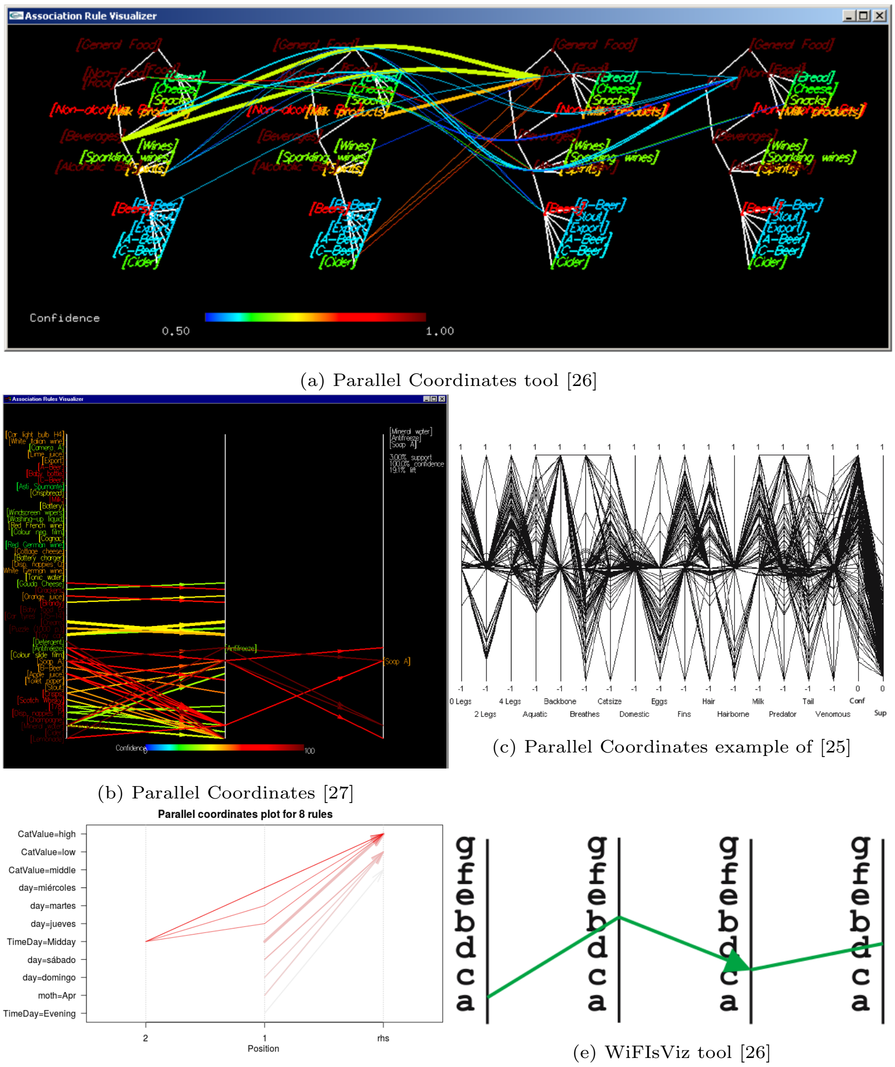
However, the major disadvantage of tabular representations is that they are not suitable for interpreting and understanding a large set of rules, as it is very difficult to observe all the rules and relationships between sets of items at the same time. Additionally, it is complex to see from a more general point of view the relationship between items through the rules or, for example, if there are groups of independent rules. So for this kind of observations we would need to use another kind of representations.

### *3.1.2. Parallel Coordinates visualization*

Another common representation for representing rules is by means of parallel coordinates [21, 22, 23]. These were originally used to display relational records with an equal number of attributes. However, it can be also used to display data with variable lengths, such as frequent item sets and association rules.

Some examples of such tools are found in [24, 25, 26] and can be seen in Figure 3. It can be observed in Figures 3a and 3d, the tool developed in [27, 24, 17]; it uses parallel coordinates to represent rules. The representation is based on axes (horizontal) with the items and a line (vertical) for each association rule. This line crosses the axes by the items/itemsets contained in the association rule. As we can see, it is a very descriptive and intuitive technique but when the number of rules increases it becomes complex to understand. Also if the number of items increases it is quite difficult to distinguish rules that contain many items due to the amount of edges and connections generated (see Figure 3c). On the other hand, for smaller sets this kind of visualization works with very good results as we can see in Figures 3d and 3e. We can conclude as in the previous group, that they are a good option to combine with other methods and visualize a specific set of rules.

Hence, the main advantage of this type of methods is the simplicity and the ease of observing the relationship between the items. It is also very intuitive to understand how it works for small size sets of rules by a person without previous knowledge about the visualization tool.



(a) Parallel Coordinates tool [26]

(b) Parallel Coordinates [27]

(c) Parallel Coordinates example of [25]

(d) Paracoord plot [17]

(e) WiFIsViz tool [26]

Figure 3: Parallel Coordinates

However these visualization methods become difficult to understand and interpret as the number of rules or itemsets grows. This is due to the fact that the representation of some rules is overwritten with others which complicates to determine which rule is each one and which item belongs to the rule. In addition, we can also encounter problems when trying to visualize itemsets or rules with many items. In such cases, the problem is with following in the graph the long lines connecting items to determine which of them belong to the rule.

On the other hand, when there are rules with many items a great amount of lines have to be be drawn what makes the comprehension and representation of the rule quite difficult.

### *3.1.3. Matrix-based visualization*

In the literature we can find different tools that allow to make the visualization through the use of matrices (2 or 3-dimensional). In Figure 4 we can see different different matrix-based visualizations using a scatter plot [28], a 2D-matrix [20] or a 3D [17, 29].

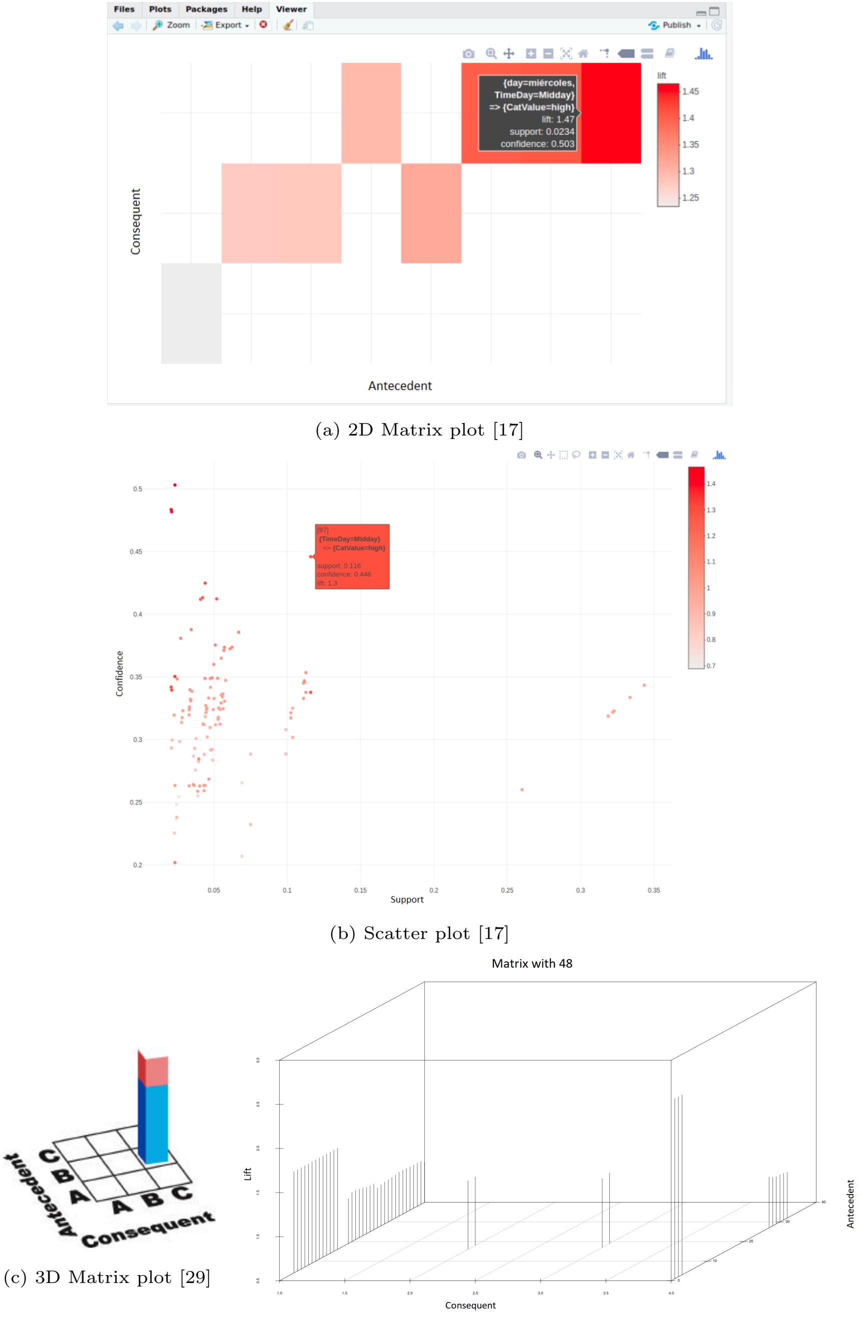
In these techniques the information to be visualized can be organized in two ways. Firstly, the rules can be displayed by antecedent and consequent in one of the axis (Figures 4a, 4c, 4d) and the measure of interest can be shown at the intersection of the antecedent and the consequent of rules. At this point there are different ways to represent it: using a third dimension and visualize the measurement with, for example, a bar graph (Figures 4c, 4d) or by color intensity (Figures 4a, 4b). Some of these tools (e.g. matrix plot in [20]) have an interactive component and by positioning the mouse in the intersection that represents a rule the user can see the different measures related to that rule.

Another way is to display only the measures (e.g. support and confidence) in the axes and to use an interactive box [30] to know which rules corresponds to that pair of values. An example is shown in Figure 4b. In this case, if our interest is to explore the obtained rules by different assessment measures, this type of visualization can be a good option.

When matrix-based visualizations use 3 dimensions, it allows a more general visualization of the set of results, helping to know what are the strongest rules. These tools represent the rules in the following way: in the axes of the matrix the items belonging to the consequent and to the antecedent are represented. In the intersection, a bar is drawn representing the measures of interest.

One of the main advantages is that we can easily know which sets of items are more frequent, or which sets of items appear in more rules. In addition, these methods show this information along with the measures of interest to improve the understanding of the results. These techniques can be customized in different ways using different colors, shapes or lengths. Also, if we want to see the whole set of rules in a general way it is a suitable method because it allows to see the related rules through the items and joint with the value of their measures of interest. For this, the best option is the 3D representation to observe easily and intuitively where the rules with the best measures are grouped.

However, this type of display has some disadvantages. First of all, it is very difficult to see the relationships between elements and rules when the set of rules/items is large. Because when the set of items is very large, the axes of the matrix are heavily subdivided in such a way that it is impossible to differentiate which itemset corresponds to each rule. Similarly, if the number of rules is very large the cells become very small and it is very complex to distinguish them.



(a) 2D Matrix plot [17]

(b) Scatter plot [17]

(c) 3D Matrix plot [29]

(d) 3D Matrix plot [17]

Figure 4: Matrix-based examples

### *3.1.4. Grouped Matrix-based visualization*

In this section, we explain a mixed technique named the grouped matrix-based visualization. In the previous case, matrix-based visualization tools were limited to the number of rules since they present difficulties when the number of items or rules increases [31]. Using the grouped version the relation between rules and itemsets can be better represented under these circumstances. In Figure 5 we can see an example of this type of visualization using the ArulesViz package [32].

Using this technique the representation of rule sets with many items in matrices is improved because the tool groups the items. This technique is based on a representation in matrices where in one axis we have itemsets of the antecedents and in the other, consequent items; but the represented rules can only have consequents with only one item. In this representation, the rules are grouped depending on the items of its antecedent, so for the same antecedent it can be shown several circles aligned with their respective consequents.

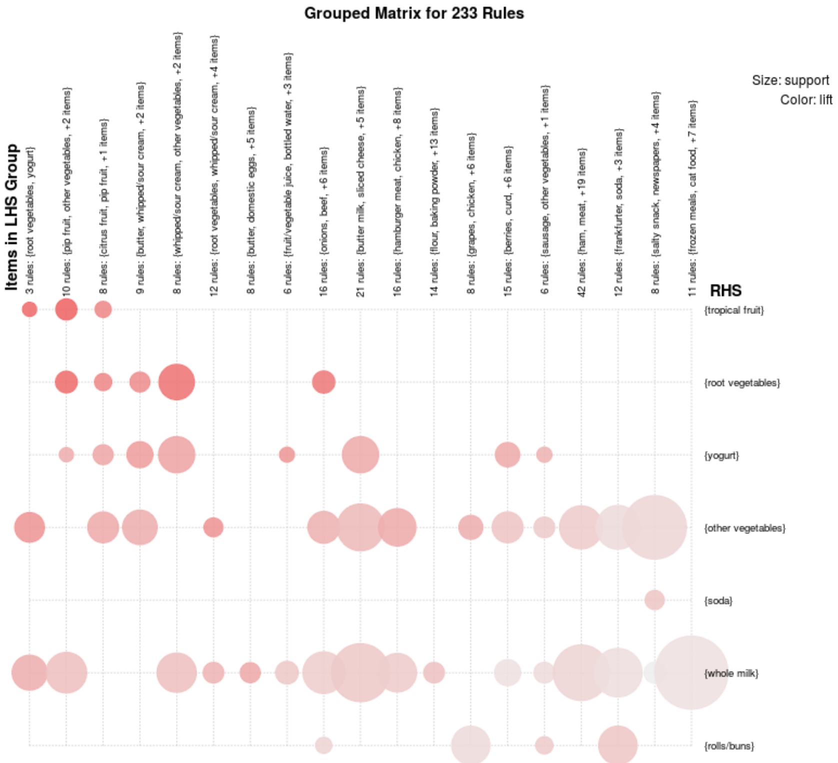


Figure 5: Grouped matrix-based visualization by ArulesViz [32]

Its main advantage is that it allows the visualization of large sets of rules. It is a good technique for analyzing rules by antecedent, since rules that have the same antecedent can be grouped together. It is also an optimal representation to relate itemsets with the different classes, since it enables to see the itemsets related with the same consequent.

Although it is a useful technique, its main weakness is that it cannot be easily adapted to display rules with more than one consequent. In addition to that, this technique has problems when visualizing very frequent itemsets due to the size of the circles. In this situation this kind of tools are not very useful since very frequent rules filled almost all the available space.

### *3.1.5. Graph-based visualization*

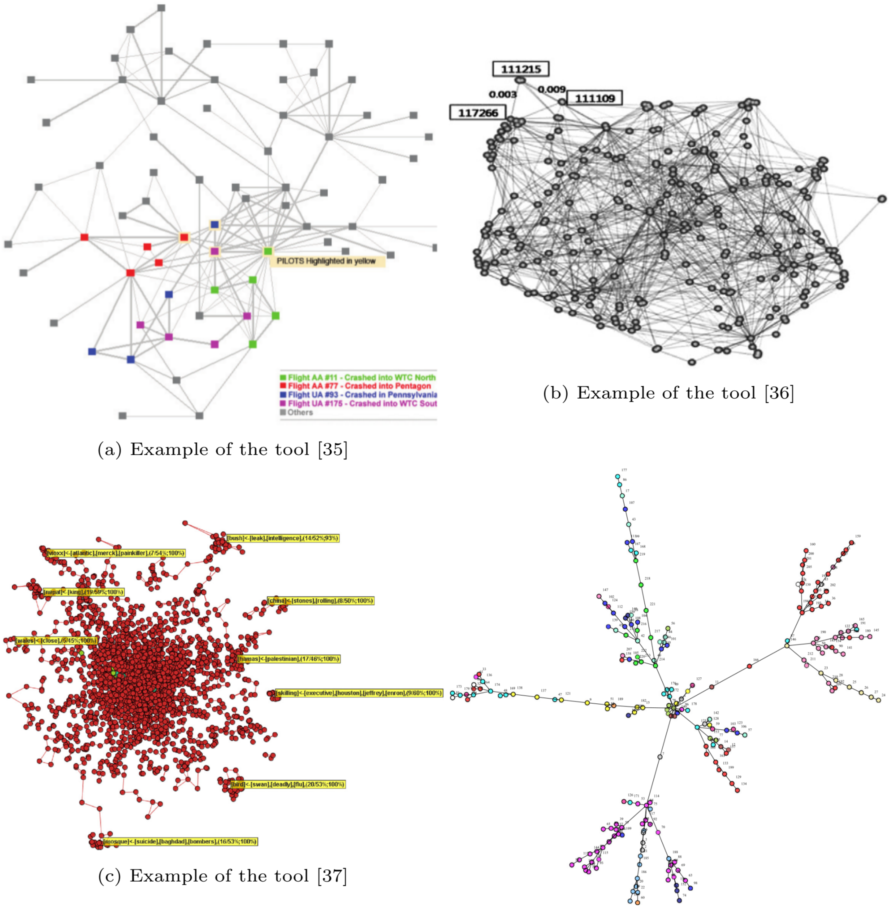
The graph-based displays consist of a transformation of the object to a structure of nodes and edges that connect the nodes. In addition, these elements can be associated with features to help the representation of more information within the network structure. In the literature we can find different ways to transform rules into graphs [33, 34]. One way is to represent each item as a node and for each rule adding an edge connecting the items belonging to the rule. Additional features can be added, such as the color of the nodes, shapes, etc.

In the literature there are different tools that make use of graphs to represent rules. Tools like [35, 36, 37, 38] allow a visual representation of the rules through some transformation into a graph. It can be observed in Figure 6, that with these tools it is difficult to observe and intuitively understand the rules. But it is a good way to observe the general set of results. Also, as can be seen in Figures 6b and 6c, it is possible to represent large sets of rules. These tools can be enhanced with interactive capabilities, e.g. [37] (depicted in Figure 6c) where a subset can be selected or zoomed in.

There are other tools oriented to graphs that use directed graphs [17, 39]. These tools improve the interpretability and compression of the visualization because through the use of directed edges we can know which node is the consequent or the antecedent (these tools are depicted in Figure 7). In them we can observe that we can also represent measures of interest of the rules by means of the forms of the nodes and edges. For example, in Figure 7a the thickens of the edges and the size of the nodes are used, and in Figure 7b the color intensity and the size of the nodes are employed to represent the confidence and the lift assessment measures.

As a result of the use of a graph structure, the visualization of the association rules can be improved by customizing the available design. Increasing thus the interoperability for visualizing large sets of rules. In addition, the graphical representation is suitable for visualizing rules with various items in the antecedent and the consequent. On the other hand it allows a more general vision of the set of rules by means of the size, color and form of the nodes and edges, as it happened for the matrix-based tools, for a better inspection of the rules, including assessment measures and the itemsets of the rules.

Nevertheless, graph-based visualizations have some drawbacks when working with large sets because, depending on the layout used, some rules are lost behind other rules. In addition, this type of visualization is not prepared to handle different measures at the same time, since using more than two layouts (e.g.


PILOTS Highlighted in yellow
🟩 Flight AA #11 - Crashed into WTC North
🟥 Flight AA #77 - Crashed into Pentagon
🟦 Flight UA #93 - Crashed in Pennsylvania
🟪 Flight UA #175 - Crashed into WTC Sout
⬛ Others
111215
0.003
0.009
111109
117266
[bush]<-[leak],[intelligence],(14/52%;93%)
[vioxx]<-[atlantic],[merck],[painkiller],(7/54%;100%)
[nepal]<-[king],(19/59%;100%)
[china]<-[stones],[rolling],(8/50%;100%)
[wales]<-[close],(5/45%;100%)
[hamas]<-[palestinian],(17/46%;100%)
[skilling]<-[executive],[houston],[jeffrey],[enron],(9/60%;100%)
[bird]<-[swan],[deadly],[flu],(20/53%;100%)
[mosque]<-[suicide],[baghdad],[bombers],(16/53%;100%)

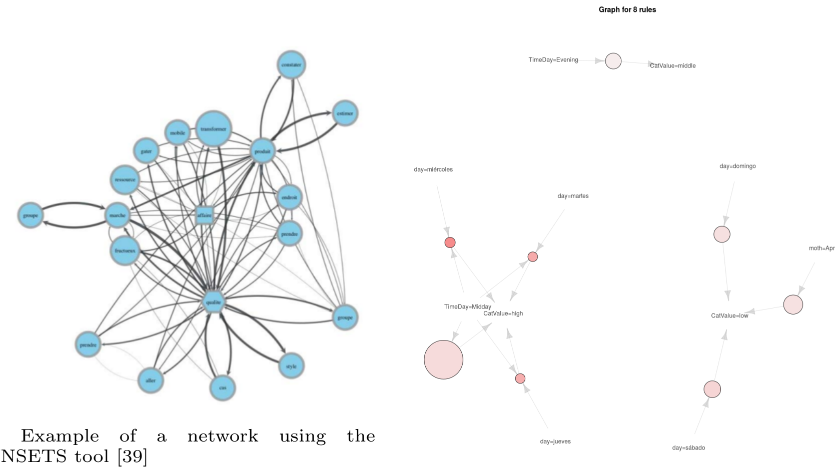
(a) Example of the tool [35]

(b) Example of the tool [36]

(c) Example of the tool [37]

(d) Example of the tool [38]

Figure 6: Graph-based examples



(a) Example of a network using the SYNSETS tool [39]

(b) Example of a graph using the ArulesViz library [32]

Figure 7: Directed Graph-based examples

size and color intensity) to represent them may cause some confusion to the user. Besides, these methods usually do not organize the items in a correct way so it is more complex for the user to see the whole set of rules and interpret them. The use of interactive tools allowing the management of the graph, as well as, filters application can greatly improve the inspection of rules represented by graphs.

### *3.1.6. Hybrid approach*

In this last group we can distinguish different approaches to visualize rules using hybrid methods. They base their functioning on the use of different visualization methods in the same tool trying to strengthen their advantages and reduce their drawbacks.

The proposal made in [29] (depicted in Figure 8a), uses a matrix to represent the rules and a bar diagram representing the support and confidence of each rule. The objective of this visualization is to solve some disadvantages of matrix-based and grid-based tools. To do this, they combine a two-dimensional matrix and a grid to create a summarized view for a quick identification of the rules according to their levels of confidence and support. The main ability of the grid is to summarize the characteristics of a set of rules in a two-dimensional plane. Combining this with a bar visualization of assessment measures in Figure 8a, it allows an analysis of rules based on the similarity of their antecedents and consequent characteristics. In this case this visualization is more like a control panel where one can select different aspects to inspect the most interesting rules. Figure 8a demonstrates the tool, specifically a 3D matrix representation for text mining. The rules are represented by a 3D box at the intersection, and depending on its color it will represent the antecedent (blue) or the consequent (red) of the rule.

Other tools, such as the ones described in [35, 40] (Figures 8b and 8c respectively), employ a dashboard to visualize, filter and select the rules to facilitate end-users the search and extraction of information. In Figure 8c we see the main interface divided in several parts: In (a) we observe a matrix with the rules and measures. In (b) we see the matrices associated to the distribution of the items contained in the rule that gives contextual information of the rule. (c) enumerates the items of interest generated in each intermediate stage. In (d) we can see the imposed thresholds. (e) contains the items selection panel. (f) enables, by means of a zoom, an interactive selection function, and (g) contains the history view which allows the user to keep the sets of interesting elements and rules from the previous step.

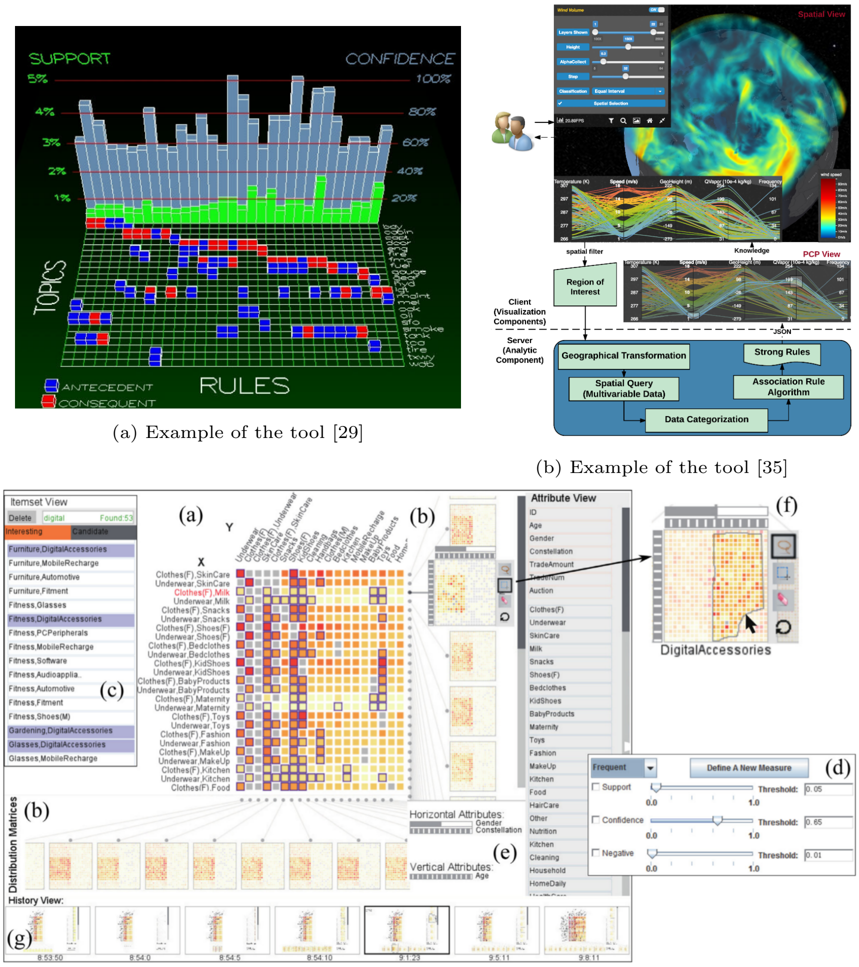
Another tool that combines different methods is [41], in this case, the authors make use of a chord diagram. We can see different examples in Figure 9. In the first of them, we can see an example with a lot of relationships, creating different chord diagrams depending on the attributes. The other two (b and c) represent sets of rules with less quantity of rules. This type of representation is very useful for appreciating the most frequent items appearing as consequent or antecedent. However, it does not have a good representation for large rule sets, although combining it with filters and other methods can be very useful, like for example its combination with a matrix or graph-based representation to select groups of rules that are displayed afterwords in the chord diagram.

### *3.2. Classification by type of capacities of the technology*

In order to visualize association rules, different types of tools can be found, depending on whether these only enable the rules visualization or to carry out the extraction process of association rules. According to this, we can classify them into two groups. On one hand, there are tools that work as libraries that allow the extraction of frequent itemsets, association rules and the visualization of the results (this process flow can be observed in Figure 10a). On the other hand, there are tools that allow loading the results obtained from other tools and visualize them (this process flow can be observed in Figure 10b).

Within the first group we find different methods that allow us to load the results obtained with association rules extraction tools. These must be loaded using a different standard structure depending on the tool for being able to visualize them. Among them we can highlight several tools [20, 25, 26, 29, 41, 42, 43] that, from the stored results as association rules, allow to visualize the rules in different ways (which are described in more detail in Section 3.1).

In the second group we find methods that enable complete analytical procedures from the same tool. For example in the Arules package [17] we can extract frequent itemsets and association rules to later visualize them making use of this library. PEAR works in a similar way, it follows the methodology developed in [44] allowing to perform the whole process, from the loading of data to the visualization. Its main difference is the use of a set of rules where

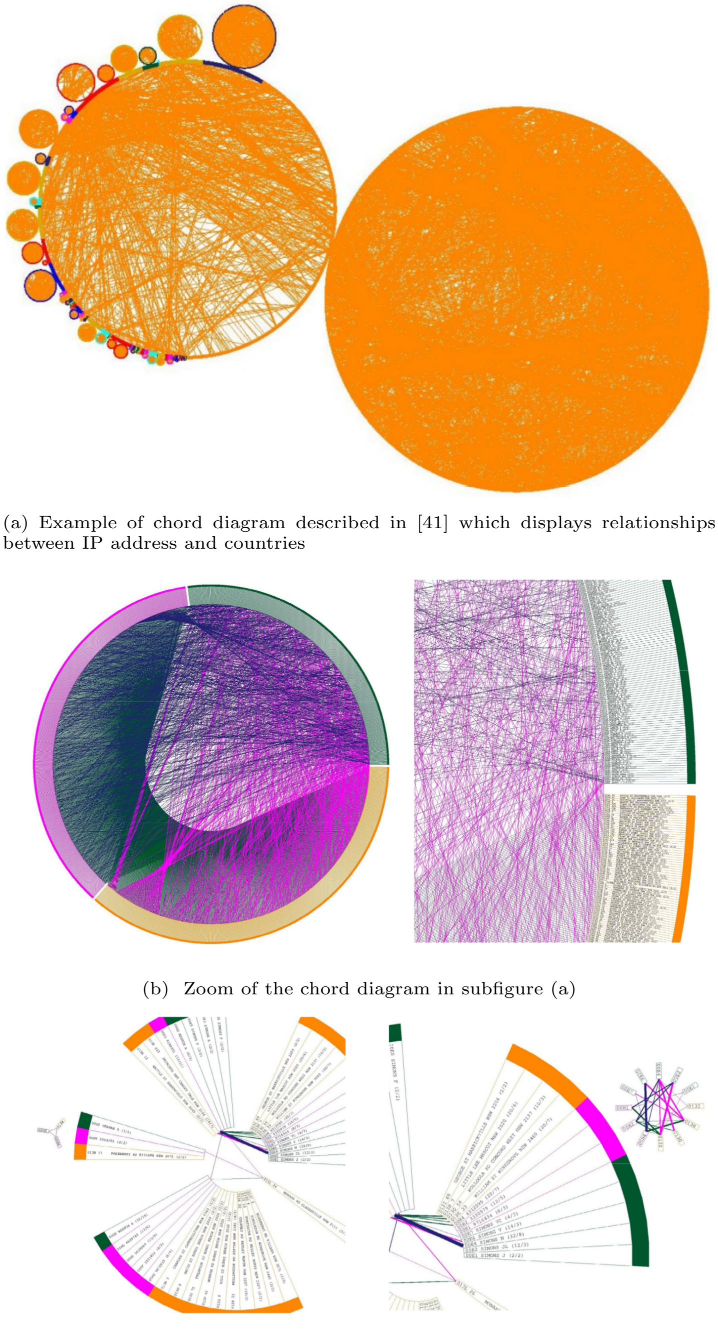


(a) Example of the tool [29]

(b) Example of the tool [35]

(c) Example of the tool [40]

Figure 8: Hybrid-based examples

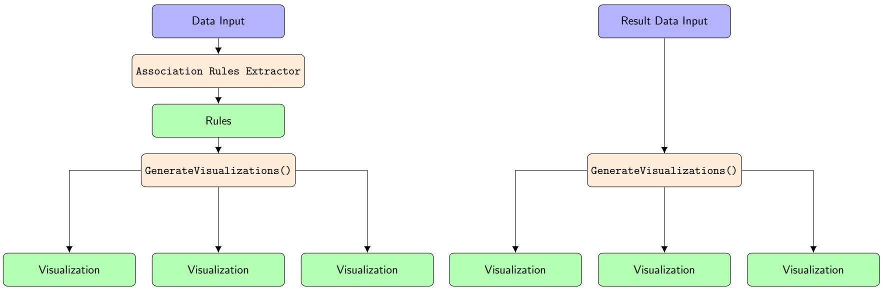


(a) Example of chord diagram described in [41] which displays relationships between IP address and countries

(b) Zoom of the chord diagram in subfigure (a)

(c) Zoom of the chord diagram in subfigure (b)

Figure 9: Hybrid-based examples



(a) Pipeline with rule extraction included

(b) Pipeline using only visualization

Figure 10: Types of pipelines employed in tools for displaying association rules

through the use of operators that allow us to unify these rules by grouping them into sets. This new set of rules is displayed with the available methods.

### *3.2.1. Comparison of tools and discussion*

All the tools reviewed in the previous sections have different advantages and disadvantages. We have summarized this information as well as their different types of functionalities in Table 1.

| Name | Type visualization | Type by capacities | Focus | Advantages | Disadvantages |
|---|---|---|---|---|---|
| Arules Table [32] | Tabular 2D | Library Methodology | Measures Rule | -Intuitive<br>-Easy to use<br>-Show a wide variety of measures | -Problematic with many rules<br>-Limited capacity with rules with many items<br>-No overall view |
| CristalClear [18] | Tabular 2D | Visualization | Measures Rule | | |
| WiFIsViz [19] | Tabular 2D | Visualization | Freq. itemsets Rules | | |
| PEAR [44] | Tabular 2D | Library Methodology | Measures Rule | | |
| Arules Scatter plot [32] | Tabular 2D | Library Methodology | Measures set of rules | | |
| Arules Matrix [32] | Matrix (2D, 3D) | Library Methodology | Rules | -Facility to explore items by measures<br>-Possibility to visualize measurements by shapes, colors | -Difficult to scan itemsets in a rule<br>-Problems when visualizing sets of many items (insufficient axis length)<br>-Overview of the whole (interactive tools improve this point) |
| Matrix [28] | Matrix (2D, 3D) | Library Methodology | Rules | | |
| Matrix for text mining [29] | Matrix (2D, 3D) | Library Methodology | Rules | | |
| Arules interactive [30] [32] Matrix | Matrix (2D, 3D) | Library Methodology | Rules | | |
| Mosaic plots [20] | Tabular 2D | Visualization | Freq. itemsets Rules | -Better compression of the relationship between items<br>-Possibility of using colors to visualize measurements | -Complexity in using rules with many items<br>-Only for reduced sets of rules<br>-No overall view |

| Name | Type visualization | Type by capacities | Focus | Advantages | Disadvantages |
|---|---|---|---|---|---|
| ArulesViz [32] | Graph | Library Methodology | Items | -Wider vision as a whole<br>-Possibility to apply different layouts<br>-Ability to view large rule sets<br>-Interactive<br>-Ability to view rules consistent with various items<br>-Use of color, shape, directionality on edges and nodes to express rule measurements | -When we work with large sets of rules are produced many nodes and edges so it is necessary to select a good algorithm of layouts for the visualization and interpretation are correct.<br>-Not prepared to handle different measurements at the same time<br>-It is more complex for the user because it is necessary to know how is the structure of a network and how are represented the rules in it.<br>-Some of them are difficult to interpret |
| VisAR [27] | Graph | Visualization | Freq. itemsets Rules | | |
| Criminal Network [35] | Graph | Visualization | Freq. itemsets Rules | | |
| Product network analysis [36] | Graph | Visualization | Freq. itemsets Rules | | |
| Projection Explorer [37] | Graph | Visualization | Freq. itemsets Rules | | |
| Spanning trees [38] | Graph | Library Methodology | Freq. itemsets Rules | | |
| SYSNETS [39] | Graph | Visualization | Freq. itemsets Rules | | |
| RDViz [45] | Graph | Library Methodology | Rules | | |
| WiFIsViz [40] [19] | Graph | Visualization | Freq. itemsets Rules | | |

| **Name** | **Type visualization** | **Type by capacities** | **Focus** | **Advantages** | **Disadvantages** |
|---|---|---|---|---|---|
| Arules Tow-key plot [32] | Tabular 2D | Library Methodology | Rule length | -Simplicity<br>-Ease to observe relationships between items<br>-Easy and intuitive understanding without knowledge | -Difficult to represent the measures of the rule<br>-Problems when increasing items or number of rules<br>-Better for rules with few related items |
| Parallel Coordinates [24] | Parallel Coordinates | Visualization | Measures Rule | | |
| Parallel Coordinates [25] | Parallel Coordinates | Visualization | Measures Rule | | |
| Parallel Coordinates [17] | Parallel Coordinates | Library Methodology | Measures Rule | | |
| Arules Grouped Matrix [32] | Grouped Matrix | Library Methodology | Measures set of rules | -Displaying Large Rule Sets<br>-Useful for rules with 1 consequent | -It is not possible to display rules with several items in the consequent<br>-Not useful for sets with very Freq. itemsets |
| Grouped Matrix [31] | Grouped Matrix | Library Methodology | Rules | | |
| CristalClear [18] | -Graph<br>Hybrid:-Chord<br>-Tree | Visualization | Measures Rule | -Observation as a whole and on concrete rules using various methods<br>-Interactive<br>-Selection and filter combining methods<br>-Diversification of measures to improve the exploration of the results | -For specific use cases<br>-More complex management tools with many filters and interactions between displays<br>-Some cases it is necessary to have certain data such as geolocation, temperature data because the displays are developed for those data. |
| Visual discovery of network patterns [41] | Hybrid:<br>-3D Matrix<br>-Bar diagram | Visualization | Freq. itemsets Rules | | |
| Hierarchical visualization [40] | Hybrid:<br>-Chord<br>-Graph | Visualization | Freq. itemsets Rules | | |
| Criminal visualization [35] | Hybrid:<br>-Chord<br>-Graph | Visualization | Freq. itemsets Rules | | |

Table 1: Comparative table of tools for visualizing association rules.

As we can see from them, there are many tools that allow to visualize association rules. However, all these techniques are independent of each other using different formats for the rules extraction and for the use of results to display them. In addition, the methods that enable the complete process from the rules extraction to the visualization have some problems because they do not allow to use other methods for visualization or extraction. That is why we propose a methodology using an intermediate form with the purpose of improving the interoperability, being able to use different visualization methods by transforming the rules into an intermediate form that can be used in a wider range of visualization tools.

On the other hand the intermediate form presented in this article allows us a format to store the results of the rules that allows to consult and efficiently filter efficiently by means of the elements and measures associated to the rules. For example, it is possible to store it natively in NoSQL databases, from which we can then display it using visualization libraries such as *bokeh, arules* or *3D.js*.

Finally, this methodology is generic because it is applicable in tools such as those that use a workflow like the one shown in Figure 10a, performing the extraction of the rules and then the visualization. Improving in this case the use not only of the visualizations that a tool implements but also being able to employ the intermediate form to use visualizations of other tools. On the other hand, in the case of the tools following the pipeline the pipeline of Figure 10b, this enables a generalization of a visualization technique for any association rule extraction algorithm

## 4. Our proposal: A new methodology using an intermediate form

We present in this work a novel methodology with the purpose of facilitating and improving the process of extraction, standardization and exploitation of the results of the existing libraries in the literature. This methodology consists in representing the rules through an intermediate form that allows the use of different tools to visualize the results as well as a more efficient structure to store them and perform searches or transformations on them.

Our proposed methodology is based on a transformation of the rules into a graph, adding different capabilities to improve their display. The procedure we followed to make a visual representation of the association (depicted in Figure 11) can be divided into three phases. The first one is in charge of obtaining the rules by means of association rules mining algorithms. Next, we transform and process these results into a graph representation. The final step is the visualization of this structure that enables the visualization of rules using different tools.

### *4.1. Association rules transformation into a graph*

The first step before visualizing the rules is to transform them into a graph; there are different options to do so. In our case, we make a transformation to a graph comprised of two types of nodes: items and rules. The nodes will be

| Antecedent | Consequent | Supp | Conf |
|---|---|---|---|
| (Other installment plans-None)<br>(Credits at this bank-1)<br>(Other debtors-none) | (Credit history-duly till now) | 0.37 | 0.7822 |
| (# persons maintained-1)<br>(Job-skilled)<br>(Other installment plans-None) | (Other debtors-none) | 0.427 | 0.9085 |
| (Housing-own)<br>(# persons maintained-1)<br>(Credits at this bank-1)<br>(Credit amount-0-3000) | (Credit history-duly till now) | 0.31 | 0.7944 |
| (Housing-own)<br>(Other installment plans-None)<br>(Credit history-duly till now) | (# persons maintained-1)<br>(Other debtors-none)<br>(Credits at this bank-1) | 0.22 | 0.7229 |
| (Housing-own)<br>(Credit amount-0-3000)<br>(Credit history-duly till now) | (Credits at this bank-1)<br>(# persons maintained-1) | 0.21 | 0.8104 |
| (Savings account-less100DM)<br>(Job-skilled)<br>(Foreign worker-yes) | (Other installment plans-None) | 0.206 | 0.8512 |
| (Credit amount-0-3000)<br>(Credits at this bank-1) | (Other debtors-none)<br>(Other installment plans-None) | 0.29 | 0.7430 |

Table 2: Set of association rules obtained for German-statlog database

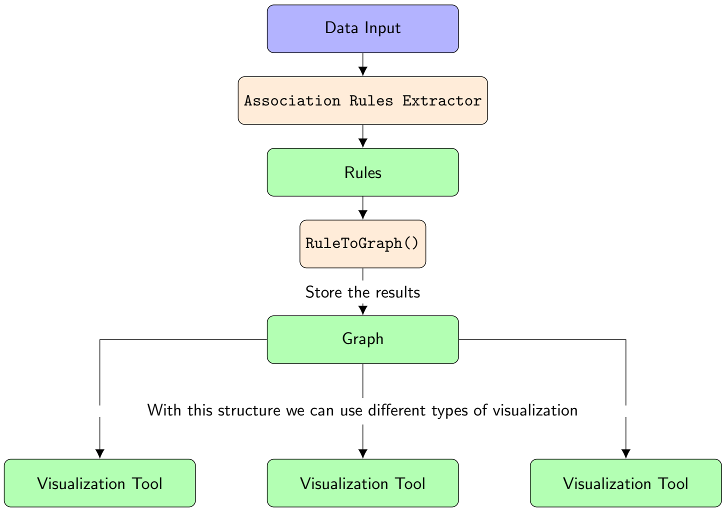


Figure 11: Pipeline proposed for association rules visualization

connected according to the results that have been obtained. To better explain the functioning of the graph transformation, we have made an example by extracting association rules from the German Credit Data (excluding continuous attributes) from the UCI Machine Learning repository [46]. We can see in Table 2 the set of selected rules that will be used in the different examples below. Figure 12 shows an example of rule transformed into a node which connects the item-type nodes.

Algorithm 1 describes the procedure for transforming a set of rules into a graph. The transformation is performed for each rule of the database, and carried out taking into account the corresponding associated characteristics according to the type of node rule of item. For example, for a rule type, the node will contain information about the measures of interest (e.g. support, confidence, lift). Then, in line 14 the edges of the graph are added, taking into account the type of link (whether it is the consequent or an antecedent). When this process ends, the rules are represented by a graph through which we can make different types of display.

Figure 13 exemplifies the structure followed for representing an item node. In this case, the name indicates the pair 'atribute_value' that represents the item; the `group` and `kind` represent the attribute `name` of the item or whether it is a node of rule type (see Figure 13). The `group` number is used to improve the efficiency in some types of visualization such as graphs. Finally, `node id` is used for univocally identify the node.

On the other hand, in Figure 14 we can see the structure for nodes representing a rule in the graph. In this case, this type of element also stores the measures related to the rule such as the support, confidence, etc. This generated structure allows a later visualization of the obtained rules in a way that

Rule: <OIP-none (Other installment plans-none), CB-1 (Credits at this bank-1), OD-none (Other debtors-none)> → <CHD-till now (Credit history-duly till now)>

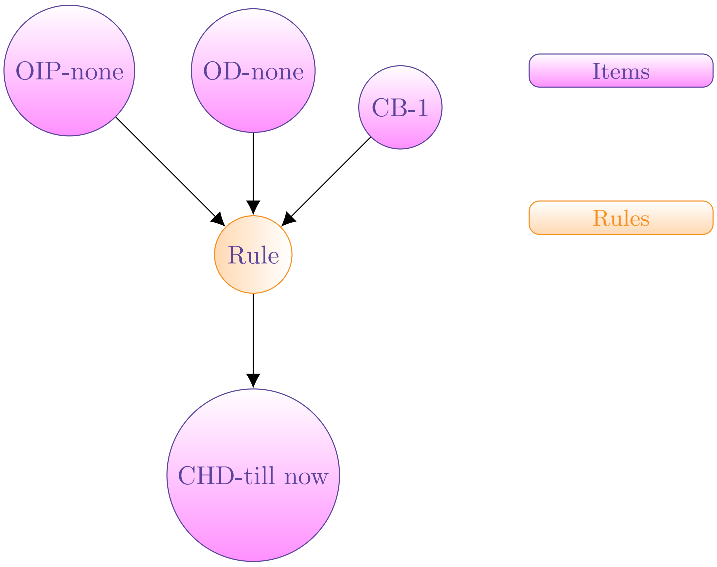
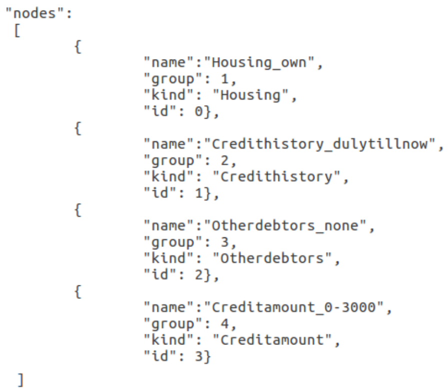
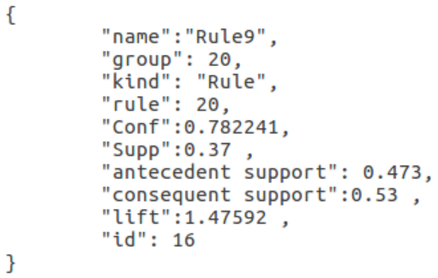


Figure 12: Example of transformation from rule to graph structure



```
"nodes":
 [
        {
                "name":"Housing_own",
                "group": 1,
                "kind": "Housing",
                "id": 0},
        {
                "name":"Credithistory_dulytillnow",
                "group": 2,
                "kind": "Credithistory",
                "id": 1},
        {
                "name":"Otherdebtors_none",
                "group": 3,
                "kind": "Otherdebtors",
                "id": 2},
        {
                "name":"Creditamount_0-3000",
                "group": 4,
                "kind": "Creditamount",
                "id": 3}
 ]
```

Figure 13: Example of Json for item type nodes



```
{
        "name":"Rule9",
        "group": 20,
        "kind": "Rule",
        "rule": 20,
        "Conf":0.782241,
        "Supp":0.37 ,
        "antecedent support": 0.473,
        "consequent support":0.53 ,
        "lift":1.47592 ,
        "id": 16
}
```

Figure 14: Example of Json for rule type nodes

improves the interpretability and comprehension of the results.

### *4.2. Intermediate form storage*

Generally, rules do not have a predefined structure for storing them that enables a fast retrieval or efficient navigation. Using the intermediate form we can store our results in different databases formats like MongoDB or Neo4j. This type of databases are specially designed for a rapid filter and sort-by order, allowing flexible and dynamic schemas.

Thanks to this structure, the rules stored following this type of JSON representation can be efficiently searched and retrieved by their measures of interest (in MongoDB by indexing the rules or in Neo4j by looking only for the type of node rule). Besides this being a standard format, exporting into databases is direct and very efficient (in MongoDB the native format is BSON a derivative format of JSON). On the other hand, if we use Neo4j we will also have the capability of querying graphs using its Cypher query language.

Finally, the intermediate form also makes it simple to query the results of

**Algorithm 1** Main transformation procedure for Rules to Graph

```
Input: Data: a csv/RDD containing the rules.
Output: Graph for association rules exceeding MinSupp
Rules = ReadData()
Graph = {}
for $n = 1$; $Number\ of\ Rules$; $n = n + 1$ do
    $Antecedent,\ Consequent \leftarrow SplitRule(Rules[n])$
    if $not(Antecedent\ in\ Graph)$ then
        $Graph.Add_{Node}(ConvertToNode(Antecedent))$
    end if
    if $not(Consequent\ in\ Graph)$ then
        $Graph.Add_{Node}(ConvertToNode(Consequent))$
    end if
    $Graph.Add_{Node}(ConvertToNode(RuleN))$
    $Graph.Add_{Edge}(RuleN, Antecedent, Consequent)$
end for
$Return \leftarrow Graph$
```

association rules using standard query APIs such as GraphQL. This natively allows querying of stored results using the intermediate form in a database.

### *4.3. Visualization tools*

Having transformed the rules to a graph structure following the methodology described in Figure 11, we can make use of the intermediate form to employ different visualization tools, some of which we have already described on in Section 3.

Taking into account the above mentioned structure, we can use different graph visualization libraries such as Gephi [47], D3JS [48], Bokeh [49], ggnet2 [50], NetworkX [51], NOESIS [52], etc. These tools return a graphical display of the transformed rules. In addition to these graphical visualization tools we can use other libraries such as Arules as we can see in Figure 15, where we have employed the set of rules from Table 2. Thanks to the use of our methodology we can visualize rules with more than one consequent (the rule extraction algorithm of the Arules package only allows rules of only one consequent) using an external algorithm [53] and using the intermediate form previously explained.

Moreover, we can use libraries and visualization tools to customize exiting visualizations tools to adjust according to user preferences. A powerful visualization package enabling this kind of modifications is 3DJS [48]. For example, in Figure 16a we have adapted a graph-based display called Force directed graph, where a different color is used for each rule, and every node of type 'Rule' shows the associated support and confidence of the rule. Additionally, the directionality of the arrows to the rule node indicates whether an item node is part of the antecedent (arrows pointing to the rule node) or part of the consequent (arrows pointing to the item node). A variation of the previous visualization is also

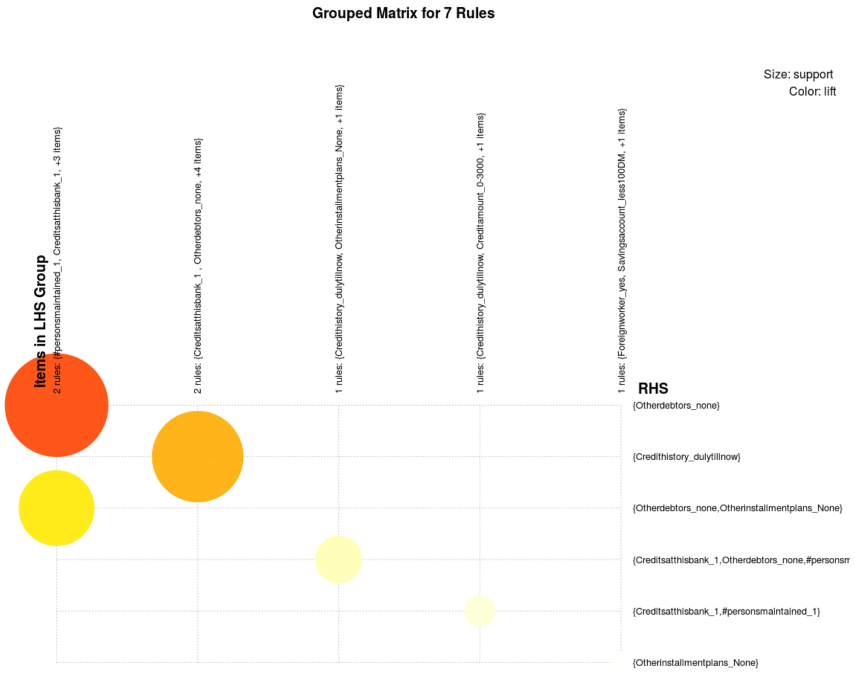


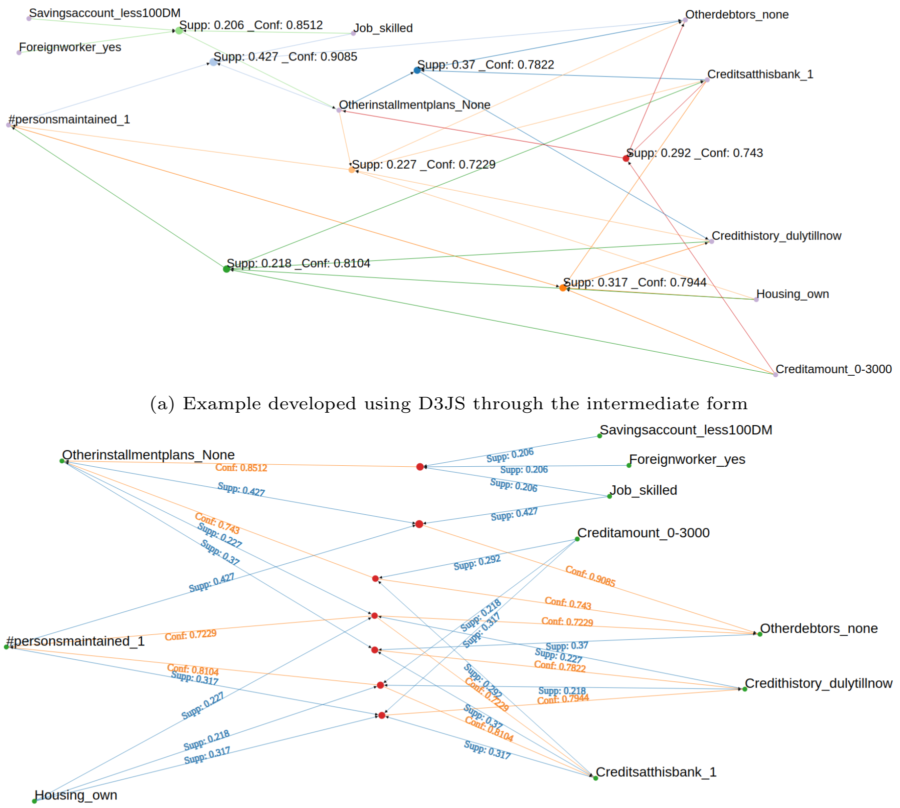
Figure 15: Association rules representation using the matrix-based visualization of [32] available in Arules library

made in Figure 16b where we have customized some properties of the graph, like for instance to indicate the item support in the edge. Another example is the chord diagram shown in Figure 17, also customized from making use of the 3DJS library. This visualization is interactive: when you are positioned in the items the edges are emphasized and colored according to the relationship they have with other items (antecedent or consequent).

With respect to the efficiency and scalability of the visualizations on graphs, the developed tool is able to generate the graphs in SVG format (Scalable Vector Graphics) being able to be distributed by means of the `Tuoris` library [54].

## 5. Lessons learned

In this paper we have reviewed the most common and used methods to visualize association rules. From the different tools examined we have seen that each one comes with strengths and weaknesses. As part of this analysis, we have been able to see how there is no one display tool that works well for all cases. This is because based on the problem, the data set and the objective, we need



(a) Example developed using D3JS through the intermediate form

(b) Example developed using D3JS through the intermediate form

Figure 16: Some display examples using the intermediate form

to choose the better visualization solution according to our requirements. In addition, some of the tools allow to export data from our own algorithms and others do not, therefore it is necessary to have a standard format with which to exchange our results among the different tools in order to exploit the strengths of each tool, adding interoperability to the process. Thus, in this way we can have more resources and not depend on a single tool to get a better visualization and understanding by the end user. Therefore our proposal allows us to extract the association rules from any algorithm or from a library and then export it to be displayed employing the different available libraries: 3D.js, Arules, Bokeh... Furthermore, this standard and widely known format can be stored natively in databases that allow to efficiently filter, search and store large sets of results.

## 6. Conclusions and future research

In summary, we have carried out an extensive analysis of the different visualization tools available in the literature, as well as classifying them according to

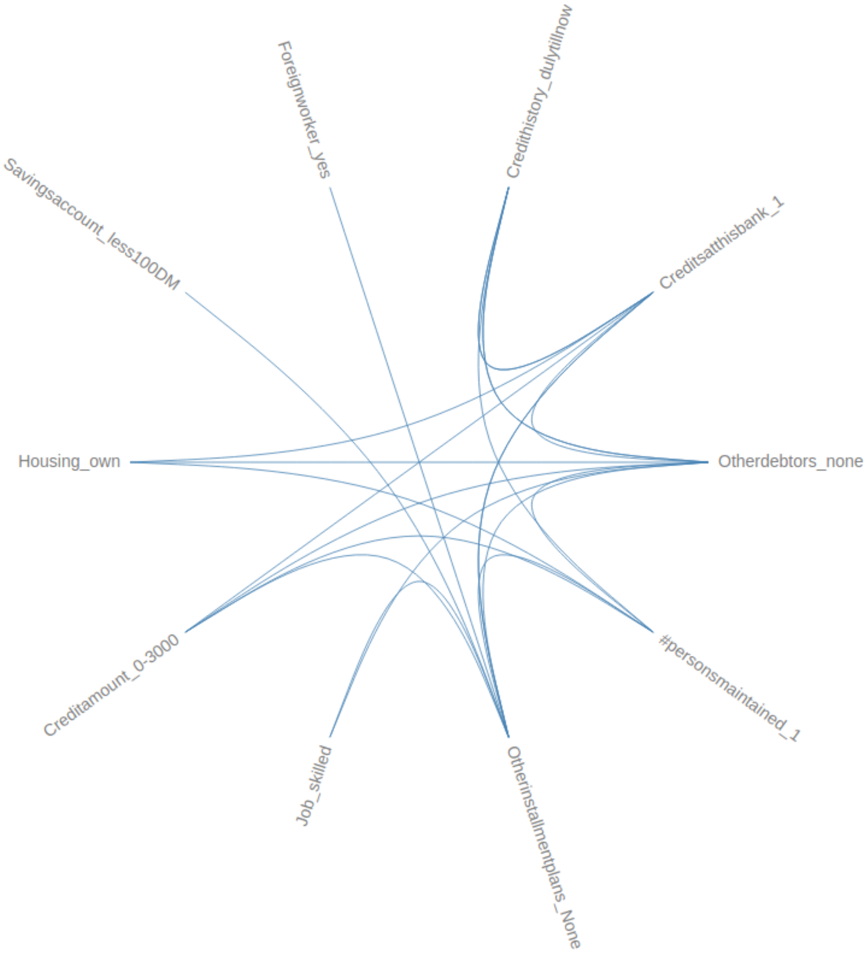


Figure 17: Example of visualization by means of a chord diagram

the type of visualization and the capabilities they offer. After this analysis we have seen that most of the tools only allow to visualize rules from a predefined structure, or the library containing the visualization only allows the generation and display of rules through their own algorithms. Therefore, in these cases, the user cannot combine their own or other available association rule mining algorithms with available techniques to present association rules in a graphical way.

Because of it, in this paper we put forward an intermediate form which allows to use a great variety of the tools for association rules, and in addition, it allows to combine them with the different available visualization techniques.

Regarding future research, we aim to implement a complete open-source library for the extraction of the rules and the transformation to the intermediate form. In addition, our purpose is to implement it in a modular way to have APIs between modules and to enable the use of different association rule extraction algorithms with the intermediate form, and facilitating the connection with available visualization libraries.

## Acknowledgments

This research paper is part of the COPKIT project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 786687, and of the project grant TIN2015-64776-C3-1-R supported by the Spanish Ministry for Economy and Competitiveness. M. Molina-Solana was supported by European Union's H2020 MSCA-IF (ga. No. 743623) and Athenea3i (ga. No. 754446) programmes.

## References

[1] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994, pp. 487–499.

[2] A. Boukerche, S. Samarah, A novel algorithm for mining association rules in wireless ad hoc sensor networks, IEEE Transactions on Parallel and Distributed Systems 19 (7) (2008) 865–877.

[3] H. Li, Y. Wang, D. Zhang, M. Zhang, E. Y. Chang, PFP: parallel fp-growth for query recommendation, in: Proceedings of the 2008 ACM conference on Recommender systems, ACM, 2008, pp. 107–114.

[4] H. Feng, M.-J. Lesot, M. Detyniecki, Using association rules to discover color-emotion relationships based on social tagging, in: R. Setchi, I. Jordanov, R. J. Howlett, L. C. Jain (Eds.), Knowledge-Based and Intelligent Information and Engineering Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 544–553.

[5] J. Xuan, X. Luo, G. Zhang, J. Lu, Z. Xu, Uncertainty analysis for the keyword system of web events, IEEE Transactions on Systems, Man, and Cybernetics: Systems 46 (6) (2016) 829–842. doi:10.1109/TSMC.2015.2470645.

[6] G. Bello-Orgaz, J. J. Jung, D. Camacho, Social big data: Recent achievements and new challenges, Information Fusion 28 (2016) 45–59. doi:10.1016/j.inffus.2015.08.005.

[7] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, M.-C. Hsu, Mining sequential patterns by pattern-growth: The prefixspan approach, Knowledge and Data Engineering, IEEE Transactions on 16 (11) (2004) 1424–1440. doi:10.1109/TKDE.2004.77.

[8] E. Hüllermeier, Association rules for expressing gradual dependencies, in: Proc. PKDD 2002 Lecture Notes in Computer Science, 2431, 2002, pp. 200–211.

[9] M. Delgado, M. Ruiz, D. Sánchez, New approaches for discovering exception and anomalous rules, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 19 (2) (2011) 361–399.

[10] A. Dattolo, M. Corbatto, Visualbib: A novel web app for supporting researchers in the creation, visualization and sharing of bibliographies, Knowledge-Based Systems 182 (2019) 104860.

[11] S. Gil, J. Bobadilla, F. Ortega, B. Zhu, Visualrs: Java framework for visualization of recommender systems information, Knowledge-Based Systems 155 (2018) 66–70.

[12] R. H. Koochaksaraei, I. R. Meneghini, V. N. Coelho, F. G. Guimarães, A new visualization method in many-objective optimization with chord diagram and angular mapping, Knowledge-Based Systems 138 (2017) 134–154.

[13] K. Li, On integrating information visualization techniques into data mining: A review, arXiv preprint arXiv:1503.00202.

[14] I. Kopanakis, B. Theodoulidis, Visual data mining modeling techniques for the visualization of mining outcomes, Journal of Visual Languages & Computing 14 (6) (2003) 543–589.

[15] R. Agrawal, T. Imielinski, A. Swami, Mining associations between sets of items in large databases, in: ACM-SIGMOD International Conference on Data, 1993, pp. 207–216.

[16] F. Berzal, I. Blanco, D. Sánchez, M.-A. Vila, A new framework to assess association rules, in: Advances in Intelligent Data Analysis, Springer, 2001, pp. 95–104.

[17] M. Hahsler, S. Chelluboina, Visualizing association rules: Introduction to the R-extension package ArulesViz, R project module (2011) 223–238.

[18] H.-H. ONG, K.-L. Ong, W.-K. Ng, E. P. LIM, Crystalclear: Active visualization of association rules, in: ICDM'02 International Workshop on Active Mining AM2002, 2002.
URL `https://ink.library.smu.edu.sg/sis_research/902`

[19] C. K.-S. Leung, P. P. Irani, C. L. Carmichael, WiFIsViz: effective visualization of frequent itemsets, in: Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on, Citeseer, 2008, pp. 875–880.

[20] H. Hofmann, A. P. J. M. Siebes, A. F. X. Wilhelm, Visualizing association rules with interactive mosaic plots, in: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '00, ACM, New York, NY, USA, 2000, pp. 227–235. doi:10.1145/347090.347133.

[21] A. Inselberg, The plane with parallel coordinates, The visual computer 1 (2) (1985) 69–91.

[22] A. Inselberg, M. Reif, T. Chomut, Convexity algorithms in parallel coordinates, Journal of the ACM (JACM) 34 (4) (1987) 765–801.

[23] A. Inselberg, Visualizing high dimensional datasets and multivariate relations (tutorial am-2), in: Tutorial notes of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2000, pp. 33–94.

[24] L. Yang, Pruning and visualizing generalized association rules in parallel coordinates, IEEE Transactions on Knowledge and Data Engineering 17 (1) (2005) 60–70. doi:10.1109/TKDE.2005.14.

[25] D. Bruzzese, C. Davino, Visual mining of association rules, in: Visual Data Mining, Springer, 2008, pp. 103–122.

[26] L. Yang, Visual exploration of frequent itemsets and association rules, in: Visual Data Mining, Springer, 2008, pp. 60–75.

[27] L. Yang, Visualizing frequent itemsets, association rules, and sequential patterns in parallel coordinates, in: International Conference on Computational Science and Its Applications, Springer, 2003, pp. 21–30.

[28] R. J. Bayardo Jr, R. Agrawal, Mining the most interesting rules, in: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, Citeseer, 1999, pp. 145–154.

[29] P. C. Wong, P. Whitney, J. Thomas, Visualizing association rules for text mining, in: Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis' 99), IEEE, 1999, pp. 120–123.

[30] C. Sievert, C. Parmer, T. Hocking, S. Chamberlain, K. Ram, M. Corvellec, P. Despouy, Plotly: Create Interactive Web Graphics, R package version 4 (1) (2017) 1.

[31] A. Unwin, H. Hofmann, K. Bernt, The twokey plot for multiple association rules control, in: European Conference on Principles of Data Mining and Knowledge Discovery, Springer, 2001, pp. 472–483.

[32] M. Hahsler, ArulesViz: interactive visualization of association rules with R, The R Journal 9 (2) (2017) 1.

[33] S.-J. Yen, A. L. P. Chen, A graph-based approach for discovering various types of association rules, IEEE Transactions on Knowledge and data Engineering 13 (5) (2001) 839–845. doi:10.1109/69.956106.

[34] W. Fan, X. Wang, Y. Wu, J. Xu, Association rules with graph patterns, Proceedings of the VLDB Endowment 8 (12) (2015) 1502–1513.

[35] J. Xu, H. Chen, Criminal network analysis and visualization, Commun. ACM 48 (6) (2005) 100–107. doi:10.1145/1064830.1064834.

[36] H. K. Kim, J. K. Kim, Q. Y. Chen, A product network analysis for extending the market basket analysis, Expert Systems with Applications 39 (8) (2012) 7403–7410. doi:10.1016/j.eswa.2012.01.066.

[37] A. A. Lopes, R. Pinho, F. V. Paulovich, R. Minghim, Visual text mining using association rules, Computers & Graphics 31 (3) (2007) 316–326. doi:10.1016/j.cag.2007.01.023.

[38] M. A. Valle, G. A. Ruz, R. Morrás, Market basket analysis: Complementing association rules with minimum spanning trees, Expert Systems with Applications 97 (2018) 146–162. doi:10.1016/j.eswa.2017.12.028.

[39] F. Simard, J. St-Pierre, I. Biskri, Mining and visualizing robust maximal association rules on highly variable textual data in entrepreneurship, in: Proceedings of the 8th International Conference on Management of Digital EcoSystems, ACM, 2016, pp. 215–222.

[40] W. Chen, C. Xie, P. Shang, Q. Peng, Visual analysis of user-driven association rule mining, Journal of Visual Languages & Computing 42 (2017) 76–85. doi:10.1016/j.jvlc.2017.08.007.

[41] S. J. Simoff, J. Galloway, Visual discovery of network patterns of interaction between attributes, in: Visual Data Mining, Springer, 2008, pp. 172–195.

[42] T. R. Gabriel, K. Thiel, M. R. Berthold, Rule visualization based on multidimensional scaling, in: 2006 IEEE International Conference on Fuzzy Systems, IEEE, 2006, pp. 66–71.

[43] M. Hahsler, S. Chelluboina, Visualizing association rules in hierarchical groups, in: 42nd Symposium on the Interface: Statistical, Machine Learning, and Visualization Algorithms (Interface 2011). The Interface Foundation of North America, Citeseer, 2011.

[44] A. Jorge, J. Poças, P. J. Azevedo, A methodology for exploring association models, in: Visual Data Mining, Springer, 2008, pp. 46–59.

[45] C. K.-S. Leung, C. L. Carmichael, FpVAT: a visual analytic tool for supporting frequent pattern mining, ACM SIGKDD Explorations Newsletter 11 (2) (2010) 39–48.

[46] D. Dua, C. Graff, UCI machine learning repository (2017).
URL http://archive.ics.uci.edu/ml

[47] M. Bastian, S. Heymann, M. Jacomy, Gephi: An open source software for exploring and manipulating networks, in: Third international AAAI conference on weblogs and social media, 2009.
URL http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154

[48] M. Bostock, V. Ogievetsky, J. Heer, D$^3$ data-driven documents, IEEE transactions on visualization and computer graphics 17 (12) (2011) 2301–2309. doi:10.1109/TVCG.2011.185.

[49] Bokeh Development Team, Bokeh: Python library for interactive visualization, http://www.bokeh.pydata.org, Last accessed on 2019-05-30 (2014).

[50] S. Tyner, F. Briatte, H. Hofmann, Network visualization with ggplot2, The R Journal.

[51] A. Hagberg, P. Swart, D. S. Chult, Exploring network structure, dynamics, and function using networkx, Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (2008).

[52] V. Martínez, F. Berzal Galiano, J. C. Cubero Talavera, The NOESIS network-oriented exploration, simulation, and induction system, CoRR abs/1611.04810. arXiv:1611.04810.
URL http://arxiv.org/abs/1611.04810

[53] C. Fernandez-Basso, M. D. Ruiz, M. J. Martin-Bautista, Extraction of association rules using big data technologies, International Journal of Design & Nature and Ecodynamics 11 (3) (2016) 178–185. doi:10.2495/DNE-V11-N3-178-185.

[54] V. Martínez, S. Fernando, M. Molina-Solana, Y. Guo, Tuoris: A middleware for visualizing dynamic graphics in scalable resolution display environments, Future Generation Computer Systems 106 (2020) 559–571. doi:10.1016/j.future.2020.01.015.

## 4.2 A comparative analysis of tools for visualizing association rules: A proposal for visualising fuzzy association rules (EUSFLAT congress)

- Carlos Fernandez Basso, M. Dolores Ruiz,M. Delgado, Maria J. Martin-Bautista.
  - Status: **Published**.
  - The 11th Conference of the European Society for Fuzzy Logic and Technology organized jointly with the IQSA Workshop on Quantum Structures Prague, September 9-13, 2019

# A comparative analysis of tools for visualizing association rules: A proposal for visualising fuzzy association rules

**Carlos Fernandez-Basso**[a] **and M. Dolores Ruiz**[a] **and Miguel Delgado**[a] **and Maria J. Martin-Bautista**[a]
[a]Computer Science and A.I. Department, CITIC-UGR, University of Granada, Spain
{cjferba,mdruiz,mdelgado,mbautis}@decsai.ugr.es

## Abstract

Discovering new trends and co-occurrences in massive data is a key step when analysing social media, data coming from sensors, etc. Although, nowadays Data Mining is very useful and widely used for the industry, business and government, the main problem of application of machine learning or data mining in other fields is the interpretability and complexity of obtained results for non-expert users in computer or data science. For this reason in the KDD process one of the most important phases is the interpretation and evaluation.

In the case of association rules is essential that results are interpretable for experts. One of the most useful tools for this goal is the visualization, because it clarifies the interpretation of results, being easier to understand in order to take a decision or explaining the behaviour of data.

**Keywords:** Association Rules, Fuzzy Association Rules, Visualization, Frequent itemsets mining, Data Mining

## 1 Introduction

Many organizations, buildings or social networks generate a large amount of data on a daily basis, such as the tweets about a particular topic, sales in Amazon, or sensors in large buildings such as for instance an airport. All of these kinds of data can be analysed by means of Data Mining techniques. Association rule mining [2] is one of the major techniques to detect and extract useful information from huge datasets such as sensors [8], recommendation systems [19] or social networks [14].

In this kind of algorithms, frequent itemset and association rule mining, are specially interesting to search relationships between events, courses and students, as for instance in MOOC courses [5]. Additionally, frequent itemsets can then be employed for discovering other types of patterns such as sequential patterns[20], gradual dependencies [17] or exception and anomalous rules[12] to discover meaningful and different patterns amongst them.

However, the nature of the data can be diverse and can be described in numerical, categorical, etc. In case of numerical variables a first approximation could be to categorise them so that, for example the value of a temperature sensor may be given by a range to which it belongs as for instance [18 C, 24 C]. However, based on how the interval is defined, the result can change. To avoid this, the use of linguistic labels such as “Warm” represented by a fuzzy set is a good option to represent the temperature of a room, having at the same time a meaningful semantic to the user.

As a consequence of the above, fuzzy association rules are an useful and important technique for extracting knowledge. Nevertheless the results obtained when applying association rule mining are, sometimes, very difficult to understand due to the large amount of rules and itemsets generated. In the literature we can find studies on revisions on tools to visualize association rule [9]. Besides, these kinds of visualizations do not allow the use of fuzzy association rules.

In this paper, we first review existing visualizations tools for frequent itemsets and association rules in the crisp and the fuzzy case, and afterwards we present several proposals to improve the visualization techniques presented when displaying fuzzy association rules.

This paper is structured as follows: in Section 2 we explain the basic concepts about association rules and fuzzy association rules. In Section 3, we study the different tools in the literature to visualize association rules. In Section 4, we propose a new way to visualize fuzzy association rules. Finally, we present some conclusions and future works.

# 2 Preliminary concepts

This section introduces all the major foundational concepts required to understand the concepts around crisp and fuzzy association rules presented in this paper.

## 2.1 Association rules

Association rules were formally defined for the first time by Agrawal et al. [1].

The problem consists in discovering implications of the form $A \to B$ where $A, B$ are subsets of items from $I = \{i_1, i_2, ..., i_m\}$ fulfilling that $A \cap B = \emptyset$ in a database formed by a set of $n$ transactions $D = \{t_1, t_2, ..., t_n\}$ each of them containing subsets of items from $I$. $A$ is usually referred as the antecedent and $B$ as the consequent of the rule.
The most commonly used measures to extract frequent itemsets and association rules are:

- The *support* [6] is the measure of the frequency with which an item appears in the database. In general, the most interesting association rules are those with a high support value.

$$Supp_D(X) = \frac{|t_i \in D : X \subseteq t_i|}{|D|} \tag{1}$$

- Given the itemsets $X$ and $Y$, and the database $D$, the confidence of the rule $X \to Y$ [6], represented as $Conf_D(X \to Y)$, is the conditional probability of $Y$ appearing in those transactions in $D$ that contain $X$.

$$Conf_D(X \to Y) = \frac{Supp_D(X \cup Y)}{Supp_D(X)} \tag{2}$$

The problem of uncovering association rules is usually developed in two steps:

- Step 1: Finding all the itemsets above the minimum support threshold. These itemsets are known as frequent itemsets.
- Step 2: Using the frequent itemsets, association rules are discovered by imposing a minimum threshold for an assessment measure such as confidence.

## 2.2 Fuzzy Association rules

To deal with uncertain and imprecise data we introduce the concept of fuzzy transaction and fuzzy association rule defined in [7, 11].

**Definition 1** *Let $I$ be a set of items. A fuzzy transaction, $t$, is a non-empty fuzzy subset of $I$ in which the membership degree of an item $i \in I$ in $t$ is represented by a number in the range [0, 1] and denoted by $t(i)$.*

By this definition a crisp transaction is a special case of fuzzy transaction. We denote by $\tilde{D}$ a database consisting in a set of fuzzy transactions.

**Definition 2** *Let $A \subseteq I$ be an itemset, i.e. a subset of items in I, the degree of membership of $A$ in a fuzzy transaction $t \in \tilde{D}$ is defined as the minimum of the membership degree of all its items:*

$$t(A) = \min_{i \in A} t(i). \tag{3}$$

**Definition 3** *Let $A, B \subseteq I$ be itemsets in the fuzzy database $\tilde{D}$. Then, a fuzzy association rule $A \to B$ is satisfied in $\tilde{D}$ if and only if $t(A) \leq t(B)\ \forall t \in \tilde{D}$, that is, the degree of satisfiability of $B$ in $\tilde{D}$ is greater than or equal to the degree of satisfiability of $A$ for all fuzzy transactions $t$ in in $\tilde{D}$ .*

The support and confidence measures are then defined using a semantic approach based on the evaluation of quantified sentences as proposed in [7, 11] using the $GD$-method [11] and the quantifier $Q_M(x) = x$,which represents the quantifier *"the majority"*.

**Definition 4** *The support of a fuzzy rule $A \to B$ is defined as:*

$$FSupp(A \to B) = \sum_{\alpha_i \in \Lambda} (\alpha_i - \alpha_{i+1}) \frac{|\{t \in \tilde{D} : t(A) \geq \alpha_i \text{ and } t(B) \geq \alpha_i\}|}{|\tilde{D}|} \tag{4}$$

*where $\Lambda = \{\alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of $\alpha$-cuts.*

**Definition 5** *The confidence of a fuzzy rule $A \to B$ is defined as:*

$$FConf(A \to B) = \sum_{\alpha_i \in \Lambda} (\alpha_i - \alpha_{i+1}) \frac{|\{t \in \tilde{D} : t(A) \geq \alpha_i \text{ and } t(B) \geq \alpha_i\}|}{|\{t \in \tilde{D} : t(A) \geq \alpha_i\}|} \tag{5}$$

*where $\Lambda = \{\alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of $\alpha$-cuts.*

# 3 Association rules visualization. Previous works and comparison

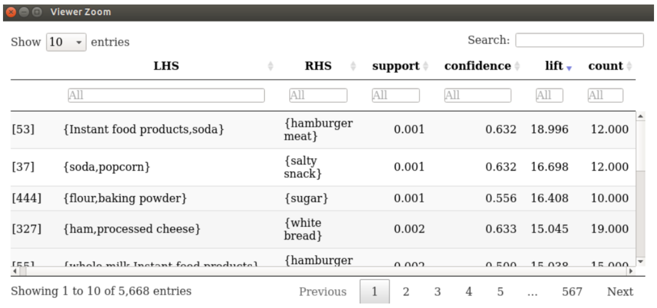
In its first subsection, basic concepts and terminology from visualization and data structure models, and its second subsection is devoted to the previous works and applications developed till the date for visualizing association rules. Afterwards, we also explain the principal advantages and drawback of these tools and we analyse the interpretability of each of them.

For visualizing association rules it can be used different ways. However we have to take into account the structure of obtained results and what we want to see in the graphic representation in order to help help the user to discover hidden insights or interpreting the results. These techniques can be classified in different subgroups so that it is clarified the different ways and structures used to represent association rules.

Consequently, we analysed the tools for visualizing association rules attending to different categories, such as data structure or kind of visualization. To sum up, in the next sections, we divide the available visualization tools in three big groups: table representation, visualization via charts like a matrix or scatter plots and graph-based methods.

### 3.1 Table visualization

Firstly, one simple method for visualizing association rules is to use a table. Although it is a simple representation, it is useful for small sets of rules since it enables the expert to observed the sets of rules ordered by antecedent, consequent take into account their measurements, such as support, confidence or lift for example. In Figure 1, we can see an example of table visualization for association rules, using the ArulesViz package. In this case, the method used allows to display the rules and rank them according to their antecedent and consequent values.



| | LHS | RHS | support | confidence | lift | count |
|---|---|---|---|---|---|---|
| | All | All | All | All | All | All |
| [53] | {Instant food products,soda} | {hamburger meat} | 0.001 | 0.632 | 18.996 | 12.000 |
| [37] | {soda,popcorn} | {salty snack} | 0.001 | 0.632 | 16.698 | 12.000 |
| [444] | {flour,baking powder} | {sugar} | 0.001 | 0.556 | 16.408 | 10.000 |
| [327] | {ham,processed cheese} | {white bread} | 0.002 | 0.633 | 15.045 | 19.000 |
| [55] | {whole milk,Instant food products} | {hamburger | 0.002 | 0.500 | 15.038 | 15.000 |

Showing 1 to 10 of 5,668 entries

Figure 1: Table

*Advantages*

The main advantage of this visualization is its simplicity, it is very easy to add and remove new assessment measures and it is very intuitive for ranking by their value. In addition, it is easily adaptable for fuzzy association rules.

*Disadvantages*

The greatest disadvantage of table representation is that it is not suitable for interpreting and understanding big set of rules, because it is very difficult to observe all rules and relationships between itemsets at the same time.

### 3.2 Matrix-based visualization

In a second place, we have grouped some techniques similar to charts. In these cases the visualization approaches use different charts for representing the association rules. In Figure 2 we can see the different ways for visualizing rules using scatter plot [4] or a Matrix in 2D [16] or 3D.

In these techniques the information to be visualized can be organized in two ways. Firstly, the rules can be displayed by antecedent and consequent in one of the axis (Figure 2a, 2c). The interest measure can be shown at the intersection of the antecedent and the consequent of rules. Another way is to display only the measures (e.g. support and confidence) in the axis and to use an interactive box [21] to know which rules corresponds to that pair of values. An example is shown in Figure 2b.

In Figure 2a we can see an example of the matrix technique where we distribute the antecedents and consequents in the axes of the graph. This tool has an interactive component and by positioning ourselves in the intersection that represents a rule we can see the different measures stored for that rule.

Finally, Scatter plot allows us to visualize the rules by two measures of the rule that are represented in the axes. In this case we can determine which rules have similar values and study the set of general rules.

*Advantages*

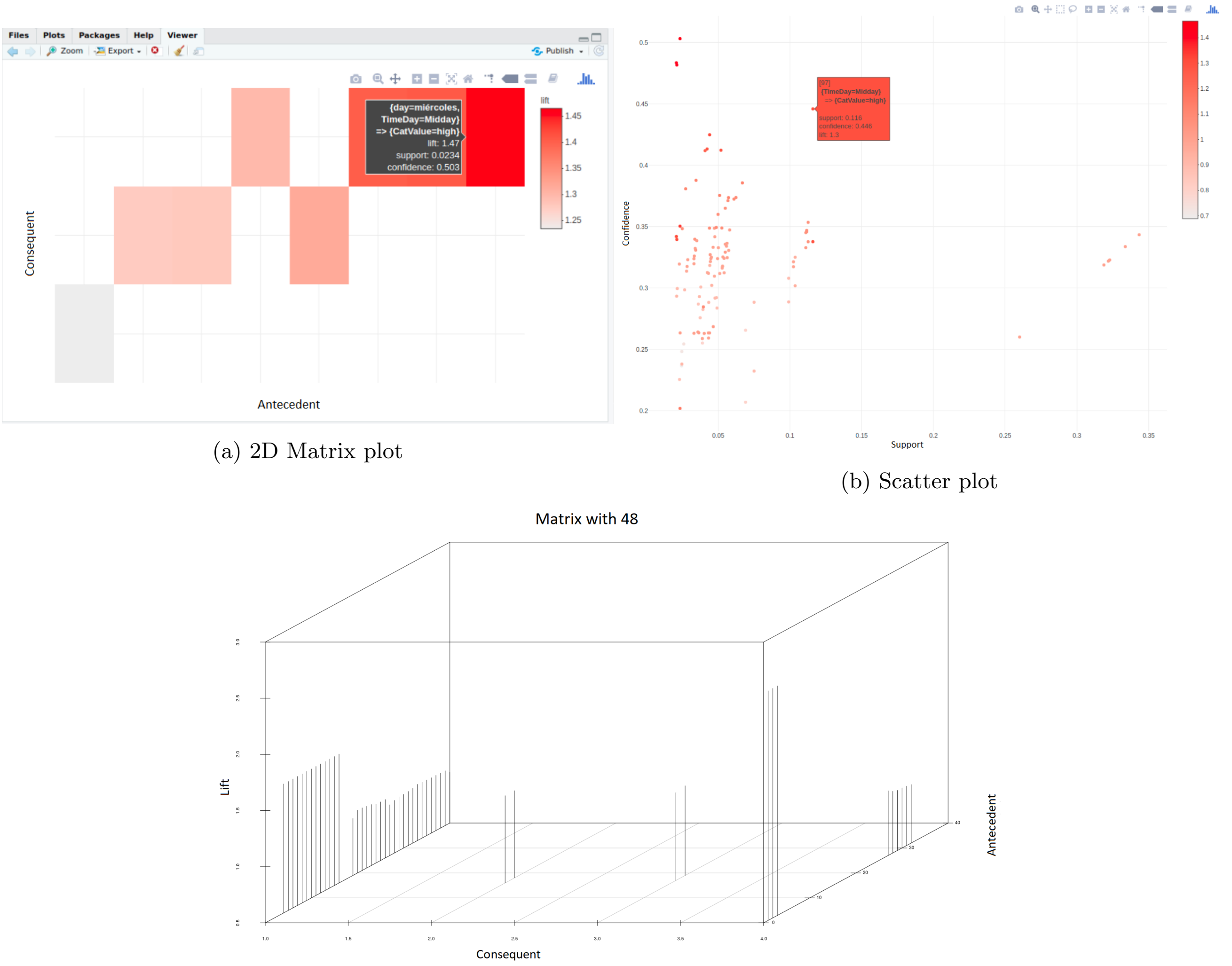
One of the main advantages is that we can easily know which itemsets are more frequent, or which itemsets appear in more rules. In addition, these methods display this information joint with the interest measures for improving the comprehension of results. These techniques can be turned in different ways using different colours, shapes or lengths.

*Disadvantages*

However, this kind of displaying have some drawbacks. Firstly, it is very difficult to see the relationships between items and rules. In addition, we can not display results obtained from large datasets because it is difficult to display a high number of items or rules with this type of charts.

### 3.3 Grouped Matrix-based visualization

In this section, we explain a mixed technique named the grouped matrix-based visualization. In the previous case, Matrix-based, were limited to the number of



(a) 2D Matrix plot

(b) Scatter plot

(c) 3D Matrix plot

Figure 2: Matrix-based examples

rules it can be visualized effectively and it presented a difficult understanding of relationships between items and rules [22]. Using the grouped version we can explain better the relation between rules and itemsets. In Figure 3 we can see an example of this type of visualization using the ArulesViz package [15].
The representation in three dimensions is similar to the one explained for two dimensions for figure 2a, but the third axis is used to visualize a measure of the rule. With this, it gains inperpretability but it is even more complex to determine which itemsets belong to a visualized rule.

*Advantages*

Its principal advantage is that it enables the display of large sets of rules. It is a good technique for analysing rule by antecedent, since rules having the same antecedent can be grouped together.

*Disadvantages*

However although it is a useful technique, its main weakness is that it cannot be easily adaptable to display rules with more than one consequent. In addition to that, this technique has some problems when visualizing very frequent itemsets. In this situation this kind of tools are not very useful since very frequent rules filled almost all the available space.

### 3.4 Graph-based visualizations

Finally, the last type of techniques are those based in a graph representation. In this way, we need to transform association rules into a graph. In the literature we can find different ways to transform rules in to graphs [24, 13]. One way is to represent each item

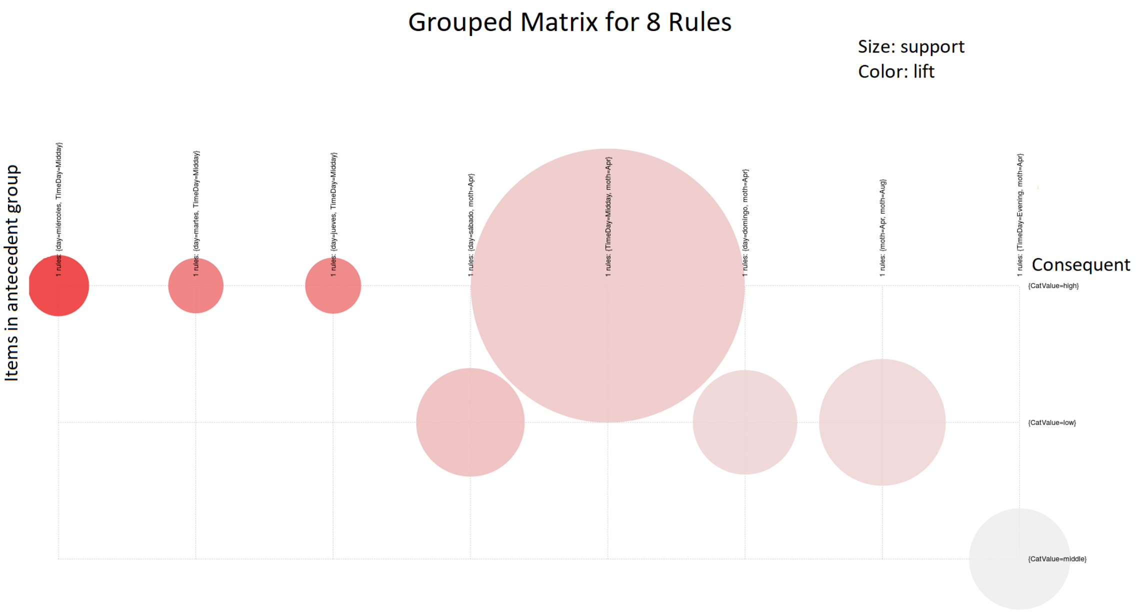


Figure 3: Grouped matrix-based visualization by ArulesViz

as a node and for each rule adding an edge connecting the items belonging to the rule.

We can observe different kinds of graphs in Figure 4. They can be divided into two groups attending to the type of layout. Within the first group we can distinguish the paracoord (parallel coordinates)[3] the WiFIsViz [10] and VisAR [23] (see the Figures 4c, 4b, 4d respectively).

In the case of WiFIsViz, we can see that this tool is very complex to understand because the representation divides the rule into antecedent and consequent difficulting to visualize the rule together. VisAR and Paracoord are very similar and they allow to understand very good the relation between itemsets in the rules, but the drawback is when we have a large set of rules or items. In these cases the tools do not have mechanisms to show all rules.

The second group contains graph without layouts restrictions (see Figure 4a and 4b). Different layout can be used and customized like the nodes colour, shapes, etc.

*Advantages*

As a result of using a graph structure, we can improve the visualization of association rules customizing the available layout. This can improve the representation and increase the interoperability when visualizing large sets of rules. Besides, the graph representation is suitable for visualizing rules with several consequents and antecedents.

*Disadvantages*

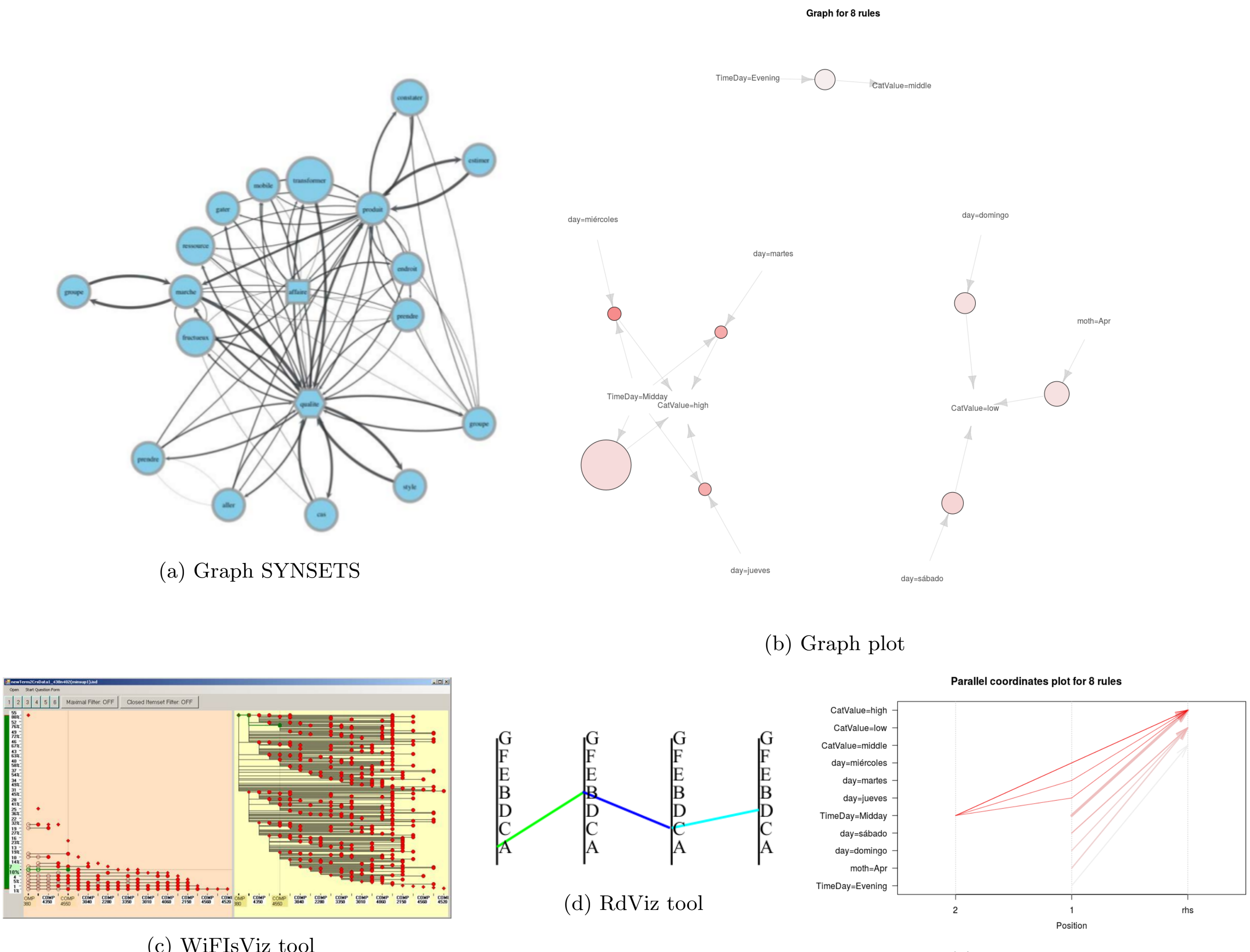
Nevertheless, the graph-based visualizations have some drawbacks when working with large sets because depending on the layout used, some rules are lost behind other rules. In addition, this type of visualization is not prepared to handle with different measures at the same time, since using more than two layouts (e.g. size and colour intensity) to represent them may cause some confussion to the user. Additionally, they are not prepared to represent fuzzy association rules.

### 3.5 Comparison of tools

In this section, we compare and analyse some characteristics of the reviewed visualization techniques. In table 1 the reader can find a summary of such characteristics.

Firstly, the basic representation as a table, is useful to observe a fragment of the complete set of rules, coming, for instance, from a query but not for a complete view for all of the discovered rules. For a broader observation of the complete set of rules it can be used the Scatter or the Matrix plot. These two methods allow, in a second step, to focus on determined rules for a detailed inspection. These visualizations are very useful to know the overall result of the mining process. In addition, it is very easy to inspect what are the main itemsets and rules together with their measures. Besides that, it can be used the grouped matrix visualization for grouping the rules by itemsets and knowing the number of rules found for each consequent and antecedent.

Lastly, it can be used the graph-based visualizations



(a) Graph SYNSETS

(b) Graph plot

(c) WiFIsViz tool

(d) RdViz tool

(e) Paracoord plot

Figure 4: Graph-based examples

| Library | Method | Type | Focus | # of measures that can be displayed | # rules |
|---|---|---|---|---|---|
| arulesViz [15] | Table | 2D | Measures and rule | * | 1000 |
| arulesViz [15] | Scatter plot | 2D | Measures and set of rules | 3(interactive information) | 1000 |
| arulesViz [15] | Tow-key plot | 2D | Rule length | 2+order | < 1000 |
| arulesViz [15] | Matrix | 2D and 3D | Antecedent and Consequent | 1(interactive information) and 1 | 100000 |
| arulesViz [15] | Grouped Matrix | 2D | Antecedent and Consequent | 2 | 1000 |
| arulesViz [15] | Graph | 2D | Items | 2 | 100 |
| FpVAT [18] | RDViz(Raw data visualization module) | 2D | Rules relationship | 3 | 1000 |
| WiFIsViz [10] | Orthogonal Graph | 2D | Rules and frequent itemsets | 3 | 1000 |
| VisAR [23] | Orthogonal Graph | 2D | Rules and frequent itemsets | 3 | 1000 |

Table 1: Comparative table of tools for visualizing association rules.

for more complex and complete displaying of rules. In this way, they can be used for more complex set of rules, enabling the visualization of different groups of rules related by itemsets. Besides, they enable to use more than two measures due to the use of different layouts configuration.

As a summary, we have found different tools for visualizing association rules, but not all of them are easily adaptable for visualizing fuzzy association rules. For this reason in next section we propose a new way for displaying fuzzy association rules.

## 4 A Proposal for Fuzzy Association Rules visualization

After the state of the art about the visualization tools for association rule made, we found that not all of the tools are suitable for displaying fuzzy association rules. For this reason we propose a new way to display this kind of rules.

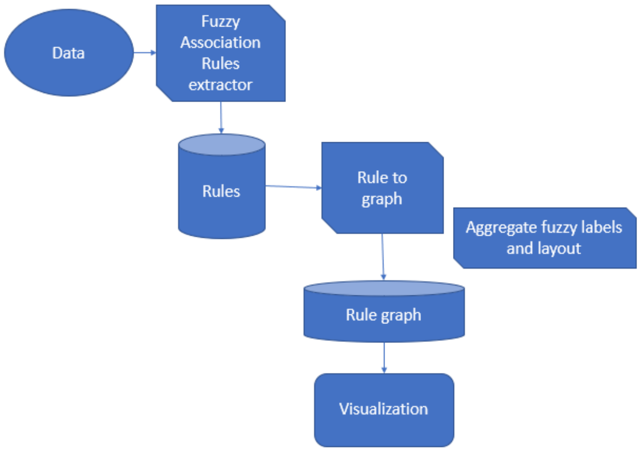

Data
Fuzzy Association Rules extractor
Rules
Rule to graph
Aggregate fuzzy labels and layout
Rule graph
Visualization

Figure 5: Pipeline proposed for fuzzy association rules visualization

The procedure (that can be seen in Figure 5) is describe in more attention to the transformation of itemsets and rules into nodes and edges respectively. Consequently, we fit the transformation by adding a new layer representing the same type of fuzzy-label as it can be seen in blue and green areas in Figure 6 like a new kind of node. Then, by means of these layers in the graph structure we can visualize the rules grouping them attending to the same type of label. For instance, if we represent the temperature, the labels "cold", "warm", and "hot" belong to the same group. We can see an example of this proposal in Figure 6, where the nodes coloured in the same cluster belong to the same type of attribute.

## 5 Conclusions and future works

To sum up we have reviewed different kinds of techniques to visualize association rules. Some of these techniques can be adapted for fuzzy association rules visualization.

The new proposal uses a Graph-Based visualization with some improvements to adapt the graph structure to represent fuzzy association rules. In this way, fuzzy association rules can be visualized by a graph, representing in addition some interesting features like the type of attributes.

As regards future research, we want to make some experimentations with large sets of fuzzy association rules. Additionally, we plan to improve the proposal made in this paper for fuzzy association rules combining different layers and/or layouts.

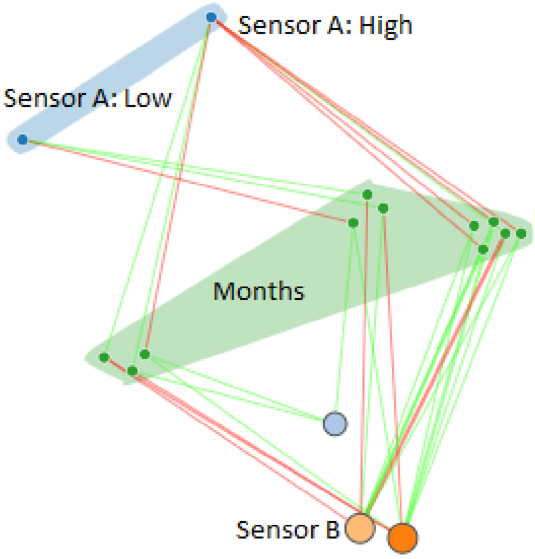


Figure 6: An example of graph visualization for fuzzy association rules.

### Acknowledgement

This research paper is part of the COPKIT project, which has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 786687 and by the Energy IN TIME project funded from the European Union under Grant agreement No. 608981, and of the project grant TIN2015-64776-C3-1-R supported by the Spanish Ministry for Economy and Competitiveness.

## References

[1] R. Agrawal, T. Imielinski, A. Swami, Mining associations between sets of items in large databases, in: ACM-SIGMOD International Conference on Data, 1993, pp. 207–216.

[2] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994, pp. 487–499.

[3] B. Almende, B. Thieurmel, T. Robert, visnetwork: network visualization using vis. js library. 2017, URL https://CRAN. R-project. org/package= visNetwork. R package version 2 (1) (2017) 172.

[4] R. J. Bayardo Jr, R. Agrawal, Mining the most interesting rules, in: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, Citeseer, 1999, pp. 145–154.

[5] G. Bello-Orgaz, J. J. Jung, D. Camacho, Social big data: Recent achievements and new challenges, Information Fusion 28 (2016) 45–59.

[6] F. Berzal, I. Blanco, D. Sánchez, M.-A. Vila, A new framework to assess association rules, in: Advances in Intelligent Data Analysis, Springer, 2001, pp. 95–104.

[7] F. Berzal, M. Delgado, D. Sánchez, M. Vila, Measuring accuracy and interest of association rules: A new framework, Intelligent Data Analysis 6 (3) (2002) 221–235.

[8] A. Boukerche, S. Samarah, A novel algorithm for mining association rules in wireless ad hoc sensor networks, IEEE Transactions on Parallel and Distributed Systems 19 (7) (2008) 865–877.

[9] W. Castillo-Rojas, A. Peralta, C. Vargas, Visualizacin exploratoria e interactiva de modelos de reglas de asociacin, Ingeniare. Revista chilena de ingeniera 23 (2015) 505–513.

[10] W. Chen, C. Xie, P. Shang, Q. Peng, Visual analysis of user-driven association rule mining, Journal of Visual Languages & Computing 42 (2017) 76–85.

[11] M. Delgado, N. Marín, D. Sánchez, M. Vila, Fuzzy association rules: General model and applications, IEEE Transactions on Fuzzy Systems 11 (2) (2003) 214–225.

[12] M. Delgado, M. Ruiz, D. Sánchez, New approaches for discovering exception and anomalous rules, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 19 (2) (2011) 361–399.

[13] W. Fan, X. Wang, Y. Wu, J. Xu, Association rules with graph patterns, Proceedings of the VLDB Endowment 8 (12) (2015) 1502–1513.

[14] H. Feng, M.-J. Lesot, M. Detyniecki, Using association rules to discover color-emotion relationships based on social tagging, in: R. Setchi, I. Jordanov, R. J. Howlett, L. C. Jain (Eds.), Knowledge-Based and Intelligent Information and Engineering Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 544–553.

[15] M. Hahsler, arulesviz: interactive visualization of association rules with r, The R Journal 9 (2) (2017) 1.

[16] H. Hofmann, A. Siebes, A. F. Wilhelm, Visualizing association rules with interactive mosaic plots, in: KDD, Vol. 2000, 2000, pp. 227–235.

[17] E. Hüllermeier, Association rules for expressing gradual dependencies, in: Proc. PKDD 2002 Lecture Notes in Computer Science, 2431, 2002, pp. 200–211.

[18] C. K.-S. Leung, C. L. Carmichael, Fpvat: a visual analytic tool for supporting frequent pattern mining, ACM SIGKDD Explorations Newsletter 11 (2) (2010) 39–48.

[19] H. Li, Y. Wang, D. Zhang, M. Zhang, E. Y. Chang, PFP: parallel fp-growth for query recommendation, in: Proceedings of the 2008 ACM conference on Recommender systems, ACM, 2008, pp. 107–114.

[20] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, M.-C. Hsu, Mining sequential patterns by pattern-growth: The prefixspan approach, Knowledge and Data Engineering, IEEE Transactions on 16 (11) (2004) 1424–1440.

[21] C. Sievert, C. Parmer, T. Hocking, S. Chamberlain, K. Ram, M. Corvellec, P. Despouy, plotly: Create interactive web graphics, R package version 4 (1) (2017) 1.

[22] A. Unwin, H. Hofmann, K. Bernt, The twokey plot for multiple association rules control, in: European Conference on Principles of Data Mining and Knowledge Discovery, Springer, 2001, pp. 472–483.

[23] L. Yang, Visualizing frequent itemsets, association rules, and sequential patterns in parallel coordinates, in: International Conference on Computational Science and Its Applications, Springer, 2003, pp. 21–30.

[24] S.-J. Yen, A. L. P. Chen, A graph-based approach for discovering various types of association rules, IEEE Transactions on Knowledge and data Engineering 13 (5) (2001) 839–845.

# 5 Applications

## 5.1 Advances in Data Science for Building Energy Management (IEECB-SC congress)

- J. Gómez-Romero , M.D. Ruiz, C. Fernández-Basso, M. Molina-Solana, M. Ros, M.J. Martin-Bautista. IEECB-SC-16 (2016).
  - Status: **Published**.
  - 9th International Conference Improving Energy Efficiency in Commercial Buildings and Smart Communities, Frankfurt, Germany, from 16 to 18 March 2016.

# Advances in Data Science for Building Energy Management

***J. Gómez-Romero***✉***, M.D. Ruiz, C. Fernández-Basso, M. Molina-Solana, M. Ros, M.J. Martin-Bautista***

***Department of Computer Science and Artificial Intelligence, University of Granada***

## Abstract

The increasing computational capabilities for information acquisition and storage have led to a massive increase of available data in different areas of interest to Energy Management; e.g. smart grid monitoring, equipment consumption measurement, user activity identification, supply and demand estimation, and building operation logging. Exploiting such *big data* offers a great opportunity to gain insights into many aspects of buildings energy performance, and therefore, to support the implementation of solid data-based policies for improving energy efficiency. Data Science comprises a set of techniques and technologies for building system models from large data volumes, with the aim of discovering and predicting trends, groups, parameter correlations, anomalies, exceptions, and other relevant patterns. Data Science has been identified as essential to address several energy efficiency challenges, such as demand prediction, operation optimization and network maintenance, to name some of them. This paper provides an introduction to the fundamentals of Data Science methods and their application to these problems. To illustrate the role Data Science in Building Energy Management, we present an illustrative example in the context of efficient building operation and maintenance.

## Introduction

The building sector is the largest energy-consuming sector, accounting for over one-third of final energy consumption globally, and an equally important source of $CO_2$ emissions, according to the International Energy Agency [1]. Around 90% of the building emissions are produced during the operational stage, primarily through the use of fossil fuels to operate the HVAC (heat, ventilation and air-conditioning) and lighting systems [2]. These figures have soared in the last decades, and they are expected to steadily increase in the near future because of the inefficiency of aging infrastructures. Therefore, the implementation of building energy saving measures has a major impact on the reduction of the contaminant emissions, as well as the most potential for delivering significant economic savings. This is the objective of the Directive 2012/27/UE of the European Parliament and of the Council [3], which establishes energy efficiency as one of the headline targets of the EU policies.

There are several complementary strategies to reduce buildings energy consumption, particularly in non-residential buildings. Sustainable architectural designs and affordable energy sources are essential in that regard, but they must be accompanied by suitable protocols to optimize energy management in order to be effective. A critical task is to adapt HVAC and lighting operation to users' needs, minimizing the equipment utilization while maintaining the occupants comfort. Usually, the actuations to reduce the energy consumption are based solely on the manager experience and informal estimations of the expected energy requirements, thus leading to inefficiencies. Nevertheless, the development and the decrease of the cost of sensor technologies in the last decade have changed this scenario. Operators have increased their awareness on their own buildings, since they are capable of monitoring them in real time and remotely applying control commands.

Now, it is time for shifting more human-based control to computer-assisted control. The large amount of data generated by the Building Management System (BMS) can be collected and exploited to obtain much more insights about the building energy behavior. Data Science –a set of techniques to discover knowledge, detect patterns, and generate predictions from large-scale data– has emerged as a suitable toolkit to this end. It offers a unique opportunity to the actors in the energy efficiency industry (including constructors, building operators and consultants) to strengthen their competitiveness and industrial leadership. So far, Data Science has been applied to address problems such as the following: (i) the prediction of energy demand in order to adapt production and distribution; (ii) the analysis of building operations as well as of equipment status and failures to optimize operation and maintenance costs; (iii) the detection of energy consumption patterns to create customized commercial offers and to detect fraud.

This paper presents an overview of different Data Science techniques, and explains how they have been employed to address these issues. Thus, in Section 2 we provide an introduction to the typical Data Science process and the most relevant techniques. In Section 3, we summarize some recent advances in the areas mentioned before. Section 4 describes an application of Data Science to a real problem in building management. Specifically, we use association rules to detect frequent situations in a test building not well solved by the current practices that may help to increase energy efficiency, both at the local and the global level. This work has been carried out in the context of the FP7 project Energy IN TIME, which aims at the development of simulation-based techniques for implementing better building automatic operation procedures. Last but not least, we finish the paper with a summary of the conclusions obtained in our research work, and some reflections about the new approaches that are expected to be predominant in the field in the next years.

## Data Science

Data Science comprises different numerical techniques aimed to automatically identify non-trivial, new, valid, potentially useful and understandable knowledge from raw data. It mostly involves mathematical and statistical data analysis supported by information technology tools. Despite this fact, the role of the human user is very important in Data Science: she is who provides expert and common-sense knowledge to guide the analysis, sets the parameter of the algorithms, and interprets the obtained results to make them actionable.

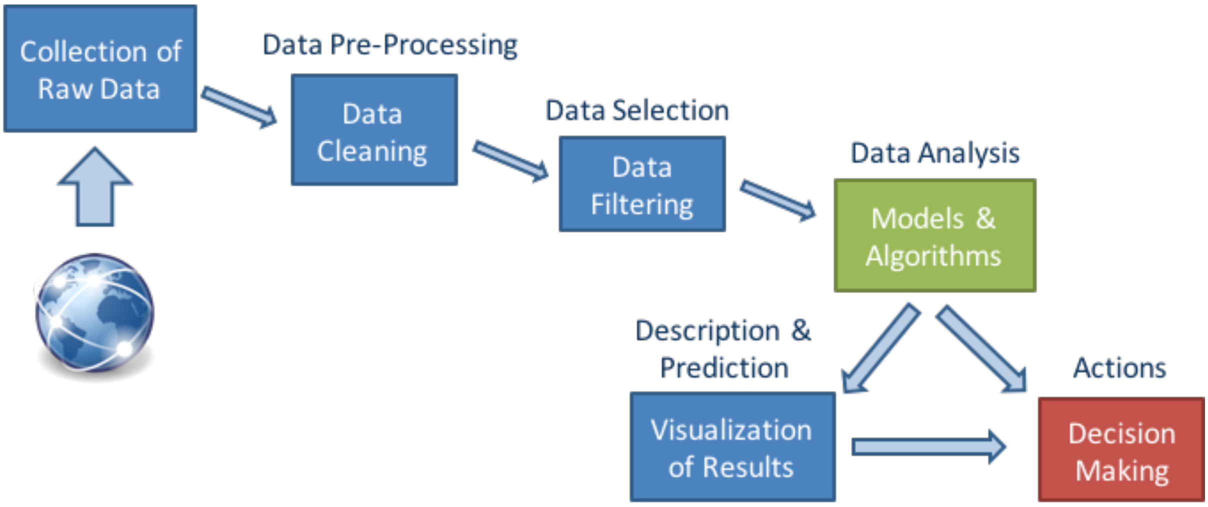


**Figure 1: Data Science process**

The typical Data Science process encompasses four steps (Figure 1): data collection and cleaning (pre-processing), filtering (selection), exploration and model building (analysis), and visualization (for data description and prediction). Here we focus on the analysis stage; for more information on pre-processing and visualization, references [4] and [5] provide, respectively, comprehensive reviews on these topics. Among the most commonly used techniques, we can find the following:

### Classification

In classification, we start from a set of objects, each defined by a collection of attributes. Attributes are represented with computable values, typically numbers. Classifying an object consists in calculating the class to which it belongs on the basis of its attributes [6]. Decision trees are a common way of performing classification: they define a kind of flowchart based on attribute values that leads us to a class. Decision trees can be automatically built (i.e. *learnt*) from a set of already classified samples according to the amount of information conveyed by an attribute; e.g. how effective is to create object partitions based on the attribute values. In general, building a data model (particularly, a decision tree) from an already classified dataset is called *supervised learning.* Other widely used classification technique is Support Vector Machines (SVM) [7], which consider objects as points in a multi-dimensional space, and calculates an optimal set of hyper-planes to separate them.

### Clustering

Clustering is the separation of objects into groups (clusters) based on an estimation of their similarity [9]. It is an unsupervised method, since there is no previous knowledge of the classes to which the objects can be assigned, or even how many of them we have. The simplest clustering methods are those based on assigning an object to the nearest cluster. The proximity is calculated by using a distance measure; e.g., the Euclidean distance of the values of their attributes. An example of distance-based clustering algorithms is k-means, which forms clusters by minimizing the mean distance between the objects inside them. A more sophisticated technique is hierarchical clustering, where clusters do not have a plain organization but rather they are arranged in cluster groups at different degrees of granularity. Clustering is often used as a first step in a classification problem when there is no training dataset with information about the classes.

### Regression

Regression analysis is the estimation of the relationship between variables [10]. Firstly, regression methods calculate if the variables are statistically correlated by means of the standard deviation, Pearson correlation, and other correlation coefficients. Secondly, a numerical model of the dependency, if any, is created. This model can be used to predict how the values of the dependent variables would change when the others do.  Regression methods can be linear, which assume that a variable can be modeled as a linear combination of other variables, or non-linear, which use more complex aggregation operators. Regression is particularly useful to predict the behavior of dynamic systems that evolve in time, since it can be applied to forecast the values of a time series based on past values. Neural Networks (NN) are also capable of building regression models. In contrast to numerical regression methods, they create black box models that are not directly interpretable, but can estimate the values of an output variable from input values. The NN models are tuned by using a backpropagation algorithm [8], which reconfigures the network links to minimize the error produced to interpolate the points of the training dataset.

### Association

The concept of association is similar to that of regression, because it also aims at discovering relationships between variables. However, association methods do not rely on strong numerical models, but in quantifying value co-occurrences: the more co-occurrences appear, the stronger the association between the variables is. One of the most used tools for modeling and estimating associations are association rules. Association rules have the form $A \rightarrow B$, which means that *A* and *B* appear frequently and with high reliability together. The Apriori algorithm is the most widely used technique for extracting association rules [11]. It is based on the computation of two statistical measures: the support, which measures how many times *A* and *B* appear together in the database; and the confidence, which measures how probable is having *B* provided a transaction that includes *A*.

Most of the previously mentioned techniques have a fuzzy extension that allows them to manage imprecise and uncertain data [12]. Fuzzy logic allows a non-strict representation of object membership to a set, thus avoiding the problem of hard boundaries that are often present in basic techniques. For example, fuzzy k-means can assign an object to one or more clusters with a strength degree. Fuzzy approaches also produce more user-friendly representations of the extracted knowledge, since it can be expressed in linguistic terms closer to human understanding.

## Applications

### Building operation

Data Science techniques can be applied to exploit the tremendous amount of data generated by BMS. One evident problem is to support the building operators to make optimal decisions in their daily work. In this regard, decision trees have been used to generate IF-THEN rules from datasets of recorded successful strategies [16]. Association rules can be also employed to extract hidden correlations in control variables [17]. Building operators can interpret this not so evident knowledge to improve their management practices. The influences of equipment operation variables energy into consumption measures can be as well studied by using NN. Specifically, a prediction model can be built to forecast the functioning of the system under different hypothesis [18].

A second aspect of building operation is medium-term planning in relation to the architectural elements of the building and the equipment. Detecting and correcting energy loss imply considerable savings, but the reasons are not always easy to identify. Clustering techniques have shown effective in that aim, helping managers to find outliers and overall malfunctioning [19]. Similarly, decision trees have been used for classifying basic architectural elements, such as walls and ceilings, according to their energy performance in order to provide support to building designers [20].

### Prediction of building energy loads

Energy load, or energy demand, is the amount of energy needed by the building in a certain period of time to operate. One important challenge in building management is to predict the energy load due to the HVAC sub-system. The energy required to operate the HVAC strongly depends on two factors: the internal loads, which refer to the heat produced by the building elements (equipment, people, lightning), and the external loads, which are influenced by external factors such as sun radiation and air temperature. Demand may not be uniform, and peak demands may happen when the building requires to be supplied with more electrical power than the average. These events are difficult to forecast, and produce several inconveniences, from inhabitant dissatisfaction to power outages.

Clustering and classification methods have been applied in the literature to characterize and group buildings according to their load profiles [13]. They have proved useful to make an initial estimation of the building behavior. Regression methods, in turn, have been commonly used to predict peak demands, usually in combination with other methods due to the high number of variables involved and the difficulty to build a regression model [14]. Recently, the role of the occupants' activities has been acknowledged as an important factor impacting the energy demand. This suggests that activity recognition methods could be exploited to incorporate this information into the energy load prediction methods [15].

### Analysis of electricity consumption

A pillar of any energy saving initiative is to understand how and when people use energy in the building. Therefore, there have been several proposals aimed at analyzing energy consumption data to characterize energy user profiles, and to distinguish behaviors with the highest potential for implementing new energy saving policies. Not surprisingly, classification and clustering methods have been applied in this regard, in particular to identify consumption patterns in domestic setups [22, 23].

It is particularly interesting to find among electricity consumption patterns those that correspond to non-technical losses (NTL), i.e. failures in the measurement equipment, either accidental or product of fraudulent manipulation. Traditionally, classification, regression and association discovery methods have been used to identify these scenarios [24]. In several approaches, the focus is not particularly centered on modeling frequent behaviors, but on detecting anomalous inconsistent consumption patterns [25].

## Use case: extracting association rules from buildings' big data

In the introduction, we have mentioned that energy management systems generate nowadays large datasets. The Data Science techniques presented in Section 2 need sufficient data to be effective, yet small enough to be manageable –being 'sufficient and 'small enough' quite imprecise and problem-specific terms. Big Data technologies have recently emerged to address the issues that appear when analyzing large datasets: volume, variety and velocity [26]. They allow us to reliably run Data Science algorithms in a distributed computing platform, which may even be virtual and transparent to the developers (in *the* cloud).

In a previous work, we proposed an algorithm that extracts association rules from very big datasets [21]. For illustration purposes, we have applied this algorithm to a database of sensor data collected from an intelligent building located in the center of Spain. The database contains more than 160.000 measurements (transactions) of more than 1.000 variables. The set of variables include temperature, humidity, power supplied by different electrical systems, status of cooling and heating systems, and energy consumption logs. Our objective with this analysis is to study the dependencies between the variables, and their impact to the overall energy consumption of the building.

Generally speaking, in data mining processes it is very important to engage the participation of the expert in order to elicit the most interesting relations, and particularly which of them were previously unknown, in order to implement new building operation policies. In this example we have followed a quite straightforward approach, but it is worth to mention that the expert building manager would have a more active participation in the process to provide support for tuning the algorithm, and more importantly, to interpret the discovered associations.

Before applying the rule extraction algorithm, the dataset was conveniently cleaned and pre-processed, following the workflow depicted in Figure 1. In this way, we prevent the problems that may arise in the first steps of the process, such as the management of missing values (time periods for which we do not have sensor data), or the configuration of the filtering process (to avoid losing information). Afterwards, the association rule mining algorithm was executed in a computing cluster with several combinations of parameters, thus obtaining in some cases thousands of rules.

Many of the extracted association rules corresponded to the expected correlations among variables, such as the intensity and the tension of an appliance. Even though these associations do not typically entail new information, they are useful to verify the normal functioning of the system, and to make explicit common-sense knowledge. Other correlations found between variables worth to notice are those corresponding to (sometimes unintentionally) redundant sensors.

Moreover, we found correlations between different equipment actuators; i.e., the configuration setpoints of a machine are directly dependent of those applied to another one. These dependencies helped us to identify fixed operation procedures. They can be helpful to identify inefficient routines, or conversely, to simplify the operation protocols by learning undocumented usual procedures. Other potentially relevant associations that appeared in the rule set may imply equipment faults; for example, we identified a relation between a specific smoke detector and the amperage demanded in one room. Cross-data analysis, enriched with information from the building information model (BIM), can be also very relevant to improve the whole data mining process. For example, location data can be used to know which devices are close, and therefore to study in more detail the interactions between them. Furthermore, incorporating architectural and materials data would remarkably extend the scope of our initial approach.

## Conclusions

This paper has reviewed the field of Data Science and how Data Science techniques can be applied to building energy management. Specifically, we have focused on building operation, energy load prediction, and identification of consumption patterns. Our experiments show that Big Data technologies can solve the computational problems that appear when processing of large amounts of data, which are likely to have an increasing relevance with the advent of the Internet of Things –with smart meters and appliances fully connected to the Internet. However, the applications to real-world scenarios are still scarce. In our experience, one of the most important aspects to improve is achieving a greater involvement of the building managers in the data analysis process. To do this, future research work should explore two complementary directions, namely, showing the potential of Data Science to building managers, and developing more user-friendly algorithms and tools. In this way, we expect that new approaches will be less opaque, easier to use, more customizable, and above all other features, more engaging.

## Acknowledgements

This work is partly funded by the Spanish Government (TIN2012-30939 project), the European Commission (FP7 Energy IN TIME project, grant agreement no. 608981), and the University of Granada (Programa de Proyectos para la incorporación de jóvenes doctores a nuevas líneas de investigación).

## References

[1] International Energy Agency. *Transition to sustainable buildings – Strategies and opportunities to 2050* (2013). Can be downloaded at: : http://www.iea.org/publications/freepublications/publication/transition-to-sustainable-buildings.html.

[2] United Nations Environment Programme. *Buildings and climate change – Summary for decision-makers* (2009). Can be downloaded at: http://www.unep.org/sbci/pdfs/SBCI-BCCSummary.pdf.

[3] European Commission. *Energy efficiency – Saving energy, saving money* (2015). Can be downloaded at: http://ec.europa.eu/energy/en/topics/energy-efficiency.

[4] García S., Luengo, J. and Herrera F. *Data preprocessing in Data Mining* (2015). ISBN 978-3319102467. Can be ordered from Springer.

[5] Simon, P. *The visual organization: data visualization, big data, and the quest for better decisions* (2014). ISBN 978-1118794388. Can be ordered from Wiley.

[6] Wu X., Kumar V., Quinlan J.R., Ghosh J., Yang Q., Motoda H., McLachlan G.J., Ng A., Liu B., Yu P.S., Zhou Z.-H., Steinbach M., Hand D.J. and Steinberg D. *Top 10 algorithms in data mining*. Knowledge and Information Systems 14(1) 2008, pp. 1-37.

[7] Bennett K.P and Campbell C. *Support vector machines: Hype or hallelujah?* ACM SIGKDD Explorations Newsletter, 2(2) 2000, pp. 1-13.

[8] Kriesel D. *A Brief introduction to neural networks* (2007). Can be downloaded at: http://www.dkriesel.com.

[9] Jain A.K, Murty M.N and Flynn P.J. *Data clustering: A review*. ACM Computing Surveys 31(3) (1999), pp. 264-323.

[10] Chatterjee, S. and Hadi, A.S. *Regression Analysis by Example* (2013). Can be ordered from Wiley.

[11] Agrawal R., Imielinski, T. and Swami, A. *Mining associations between sets of items in massive databases*. Proc. of ACM-SIGMOD International Conference on Data (1993), pp. 207-216.

[12] Hüllermeier E. *Fuzzy methods in machine learning and data mining: Status and prospects*. Fuzzy Sets and Systems, 156(3) 2005, pp. 387-406.

[13] Prahastono I., King D. and Ozveren C.S. *A review of electricity load profile classification methods*. Proc. of 42nd International Universities Power Engineering Conference (2007), pp. 1187-1191.

[14] Fan C., Xiao F. and Wang S. *Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques*. Applied Energy, 127 (2014), pp. 1-10.

[15] Moreno M.V, Zamora M.A and Skarmeta A.F. *User-centric smart buildings for energy sustainable smart cities*. Transactions on Emerging Telecommunications Technologies, 25(1) (2014), pp. 41-55.

[16] May-Ostendorp P.T., Henze G.P., Rajagopalan B. and Corbin C.D. *Extraction of supervisory building control rules from model predictive control of windows in a mixed mode building*. Journal of Building Performance Simulation, 6(3) (2013), pp. 199-219.

[17] Xiao F. and Fan C. *Data mining in building automation system for improving building operational performance*. Energy and Buildings, 75 (2014), pp. 109-118.

[18] Kusiak A., Li M. and Tang F. *Modeling and optimization of HVAC energy consumption*. Applied Energy, 87(10) (2010), pp. 3092-3102.

[19] Ahmed A., Korres N.E., Ploennigs J., Elhadi H. and Menzel, K. *Mining building performance data for energy-efficient operation*. Advanced Engineering Informatics, 25(2) (2011), pp. 341-354.

[20] Kim H., Stumpf A. and Kim W. *Analysis of an energy efficient building design through data mining approach*. Automation in Construction, 20(1) (2011), pp. 37-43.

[21] Fernandez-Basso C., Ruiz M.D. and Martin-Bautista M.J. *Extraction of association rules with Big Data technologies*. Submitted in Proc. of the International Conference on Big Data. May 2016. Alacant (Spain).

[22] Verdu S.V., Garcia M.O., Senabre C., Marin A.G. and Franco F.J.G. *Classification, filtering, and identification of electrical customer load patterns through the use of self-organizing maps*. IEEE Transactions on Power Systems, 21(4) (2006), pp. 1672-1682.

[23] De Silva D., Yu X., Alahakoon D. and Holmes G. *A data mining framework for electricity consumption analysis from meter data*. IEEE Transactions on Industrial Informatics, 7(3) (2011): pp. 399-407.

[24] Kou Y., Lu C.-T., Sirwongwattana S. and Huang Y.-P. *Survey of fraud detection techniques*. Proc. of the IEEE International Conference on Networking, Sensing and Control 2004, (2), pp. 749-754.

[25] Sforna, M. *Data mining in a power company customer database*. Electric Power Systems Research, 55(3) (2000), pp. 201-209.

[26] D. Laney. *3-D data management: Controlling data volume, velocity and variety*. Technical report, Gartner, 2001.

## 5.2 A Probabilistic Algorithm for Predictive Control with Full-Complexity Models in Non-Residential Buildings (Published in IEEE ACCESS)

- Juan Gómez Romero, Carlos J. Fernández-Basso, M. Victoria Cambronero, Miguel Molina-Solana, Jesús R. Campaña, M. Dolores Ruiz, Maria J. Martin-Bautista. IEEE Access (2019).
  - Status: **Published**.
  - Impact Factor (JCR 2019): **4.098**
  - Subject Category: **Computer Science, Information System**
  - Rank: **23/155**
  - Quartile: **Q1**

Received February 19, 2019, accepted March 15, 2019, date of publication March 19, 2019, date of current version April 5, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2906311*

# A Probabilistic Algorithm for Predictive Control With Full-Complexity Models in Non-Residential Buildings

**JUAN GÓMEZ-ROMERO**[1], **CARLOS J. FERNÁNDEZ-BASSO**[1], **M. VICTORIA CAMBRONERO**[2], **MIGUEL MOLINA-SOLANA**[3], **JESÚS R. CAMPAÑA**[1], **M. DOLORES RUIZ**[1], **AND MARIA J. MARTIN-BAUTISTA**[1]

[1]Department of Computer Science and Artificial Intelligence, Universidad de Granada, 18071 Granada, Spain
[2]Acciona Ingeniería, 28108 Alcobendas, Spain
[3]Data Science Institute, Imperial College London, London SW7 2AZ, U.K.

Corresponding author: Juan Gómez-Romero (jgomez@decsai.ugr.es)

This work was supported in part by the Universidad de Granada under Grant P9-2014-ING, in part by the Spanish Ministry of Science, Innovation and Universities under Grant TIN2017-91223-EXP, in part by the Spanish Ministry of Economy and Competitiveness under Grant TIN2015-64776-C3-1-R, and in part by the European Union (Energy IN TIME EeB.NMP.2013-4), under Grant 608981.

**ABSTRACT** Despite the increasing capabilities of information technologies for data acquisition and processing, building energy management systems still require manual configuration and supervision to achieve optimal performance. Model predictive control (MPC) aims to leverage equipment control–particularly heating, ventilation, and air conditioning (HVAC)–by using a model of the building to capture its dynamic characteristics and to predict its response to alternative control scenarios. Usually, MPC approaches are based on simplified linear models, which support faster computation but also present some limitations regarding interpretability, solution diversification, and longer-term optimization. In this paper, we propose a novel MPC algorithm that uses a full-complexity grey-box simulation model to optimize HVAC operation in non-residential buildings. Our system generates hundreds of candidate operation plans, typically for the next day, and evaluates them in terms of consumption and comfort by means of a parallel simulator configured according to the expected building conditions (weather and occupancy). The system has been implemented and tested in an office building in Helsinki, both in a simulated environment and in the real building, yielding energy savings around 35% during the intermediate winter season and 20% in the whole winter season with respect to the current operation of the heating equipment.

**INDEX TERMS** Model predictive control, simulation, control, building energy management system.

## I. INTRODUCTION

Buildings account for more than one third of the worldwide primary energy consumption [1] and they are an equally important source of $CO_2$ emissions [2]. In western countries, non-residential buildings consume between 30-40% of the energy, mostly during the operational stage and by the HVAC (heating, ventilation, and air conditioning) systems [3]. These figures are expected to increase in the future due to inefficiency of aging infrastructures, impact of climate change in weather, and economic growth in China and India [4]. At the same time, technological advances offer great opportunities to achieve energy savings in new and old buildings. For the latter, the European Union issued in 2016 an update of the Directive on the Energy Performance of Buildings addressing the target of a 30% increase of energy efficiency by 2030 [5].

There are several complementary strategies to reduce energy consumption in existing buildings. Renovation works and retrofitting, making the most of affordable and clean sources, are essential, and to be effective, they must be accompanied by suitable operation protocols to optimize energy management [6]. As a matter of fact, selecting daily optimal setpoints for the HVAC equipment is estimated to lead to savings up to 35%, depending on the climate [7].

New approaches to building energy management systems (BEMS) offer interactive and real-time building monitoring and remote control, and provide support for simulation and optimization [8]–[10]. Still, a great deal of the decision-making is left to the operators, who must analyze available data, estimate energy demand, and propose control

The associate editor coordinating the review of this manuscript and approving it for publication was Mengchu Zhou.

rules to be implemented in the BEMS. Common a priori control strategies include optimized start/stop of equipment, chiller and boiler optimization, adaptive control, and optimal energy sourcing [11].

In the last decade, several proposals for automating the generation of operational plans based on Model Predictive Control (MPC) have been presented [12], [13]. MPC uses a simulation model of the building to capture its dynamic characteristics and predict its response to alternative control scenarios. It pursues a (conflicting) dual target: reducing energy consumption thanks to pre-emptive control and anticipation of the building state while keeping users' comfort. By establishing a complete sequence of instructions for the building equipment –i.e. the (daily) operational plan–, it overcomes the limitations of homeostatic controllers, which cannot guarantee long-term optimal operation: the ahead time and the timespan of the control instructions can expand to several hours, leading to plans entailing more uncertainty –because of the use of forecasted building conditions (e.g. weather, occupancy)– and more complexity –because of the exponential increase of possible plans–, but also more efficient –because of the exploitation of the inertial effects of HVAC equipment.

MPC is formulated as a combinatorial optimization problem, in which a search algorithm must find the best actuation plan, in terms of thermal comfort and overall building consumption, in a solution space including all the possible setpoint combinations for a given future period [14]. Nevertheless, most works tend to simplify the models (e.g. by reducing the model differential equations to linear combinations) or to reduce the search space (e.g. by limiting the control to a small part of the building equipment, and by incorporating manually-extracted expert knowledge). This results in short-scope, limited-extensibility and low-performance solutions involving a great deal of manual work

The departing hypothesis of our research work is that we can exploit the increasing capabilities of massive and parallel data processing technologies to run a large amount of simulations with full-complexity physical models and to assess multiple hypothetical control scenarios to obtain the appropriate setpoints in terms of efficiency and comfort. Availability of sensor data allows us to develop more accurate models, since data can be used for calibration, calculation of better predictions of relevant contextual factors (e.g. occupancy), and detection of control performance decline. At the same time, physical models are more interpretable and easier to extend; actually, we can use physical models and model development tools out of the box, such as TRNSYS, EnergyPlus or IESVE [15].

In the Energy IN TIME project,[1] we developed an advanced BEMS for optimized HVAC operation in non-residential buildings. This BEMS is powered by Big Data technologies, which provide support for massive data management for continuous model calibration, distributed execution of simulation software, accurate prediction of building conditions, and remote operation.

The core of the system is the intelligent operational plan generator (OPG) module, an MPC-like control scheduler supported by a cloud-based extension of the IESVE[2] simulation software. The OPG algorithm calculates an operational plan (OP) for a future period (typically the next day) after simulating hundreds of candidate plans under the forecasted state of the building (i.e. considering weather and occupancy estimations) in order to minimize energy consumption while guaranteeing occupants' comfort. Eventually, the OP setpoints are automatically applied to the equipment without direct involvement of the operator. To the best of our knowledge, this is the first proposal using an off-the-shell full-complexity model for predictive control.

In this paper, we describe the OPG algorithm design, implementation, and evaluation in the Sanomatalo commercial building located in Helsinki (Finland). The control strategies for this building focus on optimizing the air supply temperature setpoint and the airflow volume setpoints. The main contributions of this research work are the following:

- The OPG algorithm, based on probabilistic search, directly provides operational plans for HVAC equipment including on/off and numerical setpoint values that are directly applied through the BEMS –no additional translation from demand estimations into actions is needed.
- We extend the control horizon compared to usual MPC approaches. The OPG considers setpoints up to a 1-day period, which fits better to the usual building operation (e.g. the operator can validate control for the whole day) and offers more opportunities for longer-term energy saving policies.
- We use a full-complexity simulation model out of the box, decoupled from the optimization algorithm and directly interpretable by experts and operators. The simulation model self-recalibrates by using data directly measured from the building and runs on a cloud-based distributed version of IESVE.
- We carry out an evaluation of the system in the simulation environment and in the real building; in the latter case, over a longer period of time than related works (30 days), in line with the recommendations in [16].

Comparison with the base control, performed according to the International Performance Measurement and Verification Protocol (IPMVP) [17], yielded energy savings above 20%, with peaks above 40% at the end of the winter season.

The remainder of the paper is organized as follows. Next, we describe several related works, most of them centered in the use of simplified simulation models. In Section III, we describe the pilot building, the simulation model, and the evaluation methodology. In Section IV, we detail the design of the OPG algorithm and its features. Section V presents the

[1] The Energy IN TIME project (Simulation-based control for energy efficiency building operation and maintenance) was funded by the European Commission within the 7th Framework Programme in 2013-2017. See [66] for a brief description of the overall project results.

[2] https://www.iesve.com/VE2018

experimental setup and the results obtained in the simulation environment and in the real building compared to the baseline operation. In Section VI we discuss the contributions of our proposal in terms of energy savings and comfort achievement, as well as possible improvements to the system. Finally, we summarize the conclusions of the work and introduce prospective directions for future research.

## II. RELATED WORK

MPC was introduced by Mahdavi in 2001 [18], and was initially used offline to derive an optimized control law from sensor measurements and simulations, and to validate predefined control strategies [19], [20]. Associated small-scale experiments, most of them carried out in the simulation environment, showed that the application of MPC can effectively accomplish a reduction in energy consumption [21]. Further studies characterized and performed a preliminary evaluation of HVAC-related energy management actions that can be exploited in MPC [22]: outside air economizer cycle, programmed start and stop lead time, load reset, and occupied time adaptive control strategy. Additionally, other authors emphasized the need for considering subjective comfort measures beyond indoor temperatures and humidity thresholds, such as predicted mean vote (PMV) [23].

In contrast, current MPC-powered BEMS are not limited to only apply a plan elicited from expert knowledge and confirmed suitable after simulation. They can dynamically generate control instructions by searching an operational plan that, according to the simulation model, satisfies the expected energy demand while minimizes consumption. Nevertheless, the calculation of the fitness of a plan by simulation is computationally expensive [24].

Bianchini *et al.* [25] addressed this issue by replacing the full model of the building by a simplified linear model. The linear model is afterwards solved by using different heuristics that reduce the search to a computable mixed integer linear programming (MILP) problem. Although this solution considerably reduces the capability of the algorithm to find unknown solutions, it proved to yield good results in a simulation environment when tested for a delimited section of the building. Different proposals using linear and non-linear programming, having different degree of complexity, application scope, evaluation comprehensiveness and achieved energy savings, can be found in the literature, in particular for non-residential buildings [26]–[34].

Similarly, MPC solutions have been successfully applied to optimize the use of different energy sources in buildings with mixed supply systems [35]–[37] and to achieve distributed control [38], [39] –enabling extensions to minimize communication between network components [40]. To increase the capabilities for solution diversification, other search techniques have been applied to optimization in MPC, such as genetic algorithms [41]–[43] and particle swarm optimization [44]. To address the stabilization of the control process, nonlinear MPC solutions with varying horizon have been proposed [45].

As an alternative to MILP and related techniques, Katsigarakis *et al.* [46] created a surrogate building model by applying Machine Learning techniques. This surrogate model is automatically learnt from pre-computed outcomes of the real model by using a regression technique (e.g. support vector machines), and optimization with it is significantly faster than in MILP. Unfortunately, it can be inaccurate or unfeasible if the building state is difficult to model; i.e. when the control scope is too broad, there are too many outputs to estimate, or the variables have complex interdependencies. Analogously, Casals *et al.* used Bayesian networks to simplify the simulation model of a subway station, obtaining good prediction accuracy [47]. Their system does not provide long-term operation plans –and consequently, it does not optimize HVAC operation–, yet it achieves considerable energy savings in ventilation and lighting systems –thanks to the use of sophisticated Computer Vision techniques for real-time occupancy estimation. Manjarres *et al.* trained a predictive black-box model using Random Forests that reproduces the daily behavior of the building and replaces the physical model of the building; however, the control strategies are limited to switching on and off the HVAC systems [48]. Kontes *et al.* created a surrogate model with support vector machines (SVM) to optimize radiator operation with similar promising results [49].

A subsequent problem of MPC is the accuracy of the simulation model, particularly if a simplified version is required [50]–[52], or if there is uncertainty in the expected building conditions; e.g., weather forecast and occupancy estimations [53]–[55]. In this regard, Kwak *et al.* proposed exploiting parallel co-simulation, which is the execution of several simulation models under different conditions to minimize the errors due to uncertainty in input data and unexpected occupancy variations. The authors implemented a general-purpose enthalpy controller that generated control signals starting 15 and 30 minutes later [56], and a daily controller [57]. For the combination of the simulation models –in EnergyPlus and MATLAB–, they used the Building Controls Virtual Test Bed (BCVTB) suite. The system was tested during one day in severe weather conditions in a real building, showing energy savings around 2% in the best case.

## III. MATERIALS AND METHODS

### A. SANOMATALO BUILDING AND PILOT AREA

Sanomatalo[3] (Sanoma house, ‘house of the press’) is a multi-purpose building situated in Helsinki and inaugurated in 1999. It was designed by Jan Söderlund and Antti-Matti Siikala, featuring a double glass façade with a steel frame structure to reduce the need for heating. In its 9 floors and 8227,56 m$^2$, it houses the offices of the Sanoma media group and offers 2 floors of covered public space. The building is managed by Caverion,[4] a Finnish construction and maintenance company.

[3] https://sanoma.fi/en/sanoma-house/

[4] https://www.caverion.com/

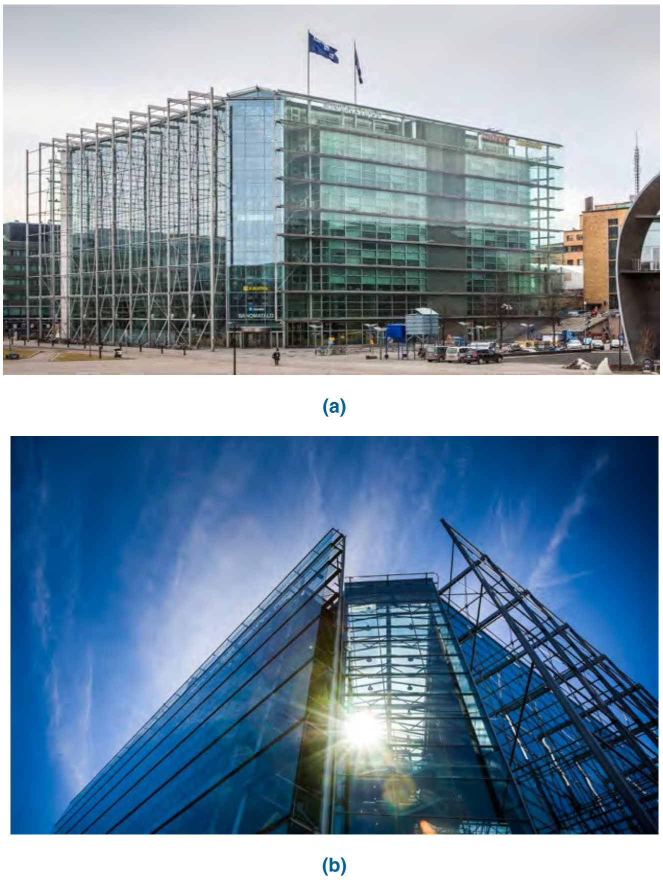


**FIGURE 1.** Sanomatalo building: (a) general view; (b) detail of the façade (source: FUNIBER for the Energy IN TIME project).

The building is connected to the district heating network and rooms are heated by waterborne radiators and fan coil units. There are four heat exchangers, one of them dedicated to the AHU heating network (power = 550 kW). All areas in the building have mechanical ventilation, which adjusts airflow based on room temperature and $CO_2$ concentration. The BEMS is provided by Schneider Electric and allows controlling ventilation, heating, and cooling sub-systems from a centralized console. It enables about 2.000 inspection points, as well as an OPC (OLE for Process Control) module that allows remote setpoint writing.

The main challenge in Sanomatalo is minimizing energy consumption (and costs) while guaranteeing comfort (indoor temperature and $CO_2$ concentration) during the heating season –usually between September and May, being the period from January to March the coldest one. Indoor temperatures can be retrieved in real-time through the BEMS, whereas $CO_2$ sensors cannot be remotely accessed –data must be downloaded offline. Heating consumption is monitored every hour by a separated sub-system. District heating prices are fixed for each season, amounting to approximately 50€/MWh in the harsh winter period (Jan-Feb), and 45€/MWh in the remainder of the winter period (Mar-May, Nov-Dec). Electricity price is about 77 and 79€/MWh, respectively. No detailed historical records of sensor measurements were available at the beginning of the project in 2013, but they were acquired in 2015-2017.

For demo purposes, we identified a pilot area of 2,748.60 m$^2$ encompassing floors 6th to 8th, which include small-size offices, meeting rooms, and open polyvalent spaces. The use of the pilot area is the expected one for an office building, with flexible working hours between 6am–18pm and an overall floor space factor of 26.2 m$^2$/person. Total electricity consumption in the pilot area in 2017 from January to April was about 60 MWh, while district heating consumption was about 35 MWh in the same area and period. These floors are served by a single not-shared air handling unit (AHU), which is configured by means of a temperature setpoint. This piece of equipment was the main parameter of the energy optimization strategies (see section III.C). In addition, we adjusted the air volume setpoint of three variable air volume (VAV) units serving 8$^{th}$ floor.

### B. SIMULATION MODEL AND CALIBRATION

The accuracy of the simulation model is a crucial aspect of MPC approaches to avoid the generation of control instructions under wrong assumptions [58]–[60]. To this aim, control-oriented models must effectively catch all the interactions between HVAC equipment (radiators, heat pumps, etc.) [61]. This is however a difficult and costly process [62].

Grey-box models have showed good performance and cost-benefit ratio [63], [64], even with relatively simple formulations and few input variables [65]. This kind of models rely on the existing corpus of expert knowledge to model thermal behaviour by using differential equations encoding the physical principles of mass, energy and momentum transfer; and they apply statistical models to tune model outputs based on historical and live data.

A canonical grey-box model –namely, the *operational* model– was created at system design time with the IESVE software by IES energy experts with the support of Caverion's building operators. IESVE comprises a series of individual components including climate, geometric modelling, solar shading, energy and carbon, lighting, airflow, thermal mass, value/cost and egress modules that are linked by a single Integrated Data Model (IDM) through a Common User Interface (CUI). By combining these modules, we can model and simulate all aspects of a building's construction, location, geometry, climate, usage, sub-systems and thermal performance.

The simulation model developed for Sanomatalo included: (a) the passive components of the building (façade, claddings, solar irradiation, etc.), created with the ModelIT and the SunCast modules; (b) the active components (anything producing or consuming electricity, especially in relation to the HVAC system), created with the ApacheHVAC, MacroFlo and Vista modules; (c) the expected building conditions (predicted occupancy and weather forecast). Simulation was performed by the ApacheSim module, which dynamically simulates the interaction between all of the active and passive elements over a selected period of time, taking into account the external influences (i.e. weather and occupancy) and the internal thermal behavior. The results of the simulation were viewed in the VistaPro module for analysis of heating and cooling loads, energy consumption, internal temperatures, thermal comfort, etc.

The details of the Sanomatalo model are not public and fall out of the scope of this paper. Nevertheless, this should not be seen as a limitation of our proposal. On the contrary, our approach is agnostic to the underlying simulation model, as far as it allows setting operational profiles as input.

The parameters of the operational model were continuously adjusted to fit live data measurements with the simulation output. Calibration was implemented as a semi-automatic procedure encompassing two iterative steps: (1) measuring the model accuracy by comparing simulation outputs with measured building data; (2) modifying model parameters to reduce model errors. In addition, IES carried out an entropy analysis to detect which parameters have the greatest influence in the model output, and therefore should be firstly modified. Overall, the calibration procedure resulted in a simulation model yielding errors below 5% [66].

### C. ENERGY OPTIMIZATION STRATEGIES

Following the Energy IN TIME terminology, control strategies specify the setpoint values allowed for each piece of actionable equipment. Strategies can denote single setpoint restrictions (e.g. setpoint variable range, frequency of change) or cross-parameter restrictions (e.g. two setpoints cannot have specific values at the same time). Besides, strategies can vary depending on the season. Energy optimization strategies are strategies enriched with heuristic information aimed at improving the energy efficiency and maintaining comfort. That is, energy optimization strategies define additional setpoint constraints that can help to reduce energy consumption (e.g. reasonable length of the pre-heating period). Energy optimization strategies can be seen as the instantiation of the Energy Management Control functions proposed in [22] for a particular building.

During the plan generation process, the operational model is cloned and reconfigured according to the forecasted occupancy and weather conditions –namely, the *independent profile variables*. As introduced in Section IV.A, the occupancy was measured as the room occupancy % from the building agenda, and the weather was a set of variables including outdoor air temperature (OAT), solar irradiance, etc.

Therefore, to run a simulation, we specify the operational input profiles –i.e. the equipment setpoint sequences to be tested in the simulation– and the independent profiles –i.e. the occupancy and the weather time series–, in order to get the predicted profiles –i.e. the value sequences for indoor temperatures, $CO_2$ concentration, and energy consumption.

Energy optimization strategies for the Sanomatalo experiments with the OPG solution encompassed:

(1) The supply temperature of the AHU in the pilot area (*Tsupply*), in the range [17, 23] °C;

(2) The airflow of 3 VAV devices ($VAVairflow_i$) in floor 8th, in the range [50, 200] l/s. The choice of selecting these 3 VAVs was the limited availability of $CO_2$ sensors at the beginning of the project: only the area affected by these 3 VAVs was monitored.

In pre-OPG operation, *Tsupply* values were manually set by the operators and *VAVairflow* values were automatically set by using presence sensors.

The comfort requirements for the new system in the heating period were the following:

- Indoor air temperature (*IAT*) must be in the range [20.5, 21.5] °C during office hours 6:00–18:00. A flexible margin in [20, 22] °C is considered acceptable. This temperature was represented by 25 output simulation variables, corresponding to 25 sensors spread across the 3 floors directly accessible through the BEMS.
- $CO_2$ concentration (*Con*) upper limit is 850 ppm during office hours. This concentration was represented by 4 output simulation variables, corresponding to 4 sensors for which there were no live measurements through the BEMS.

The target variables to optimize were the heat and the fan power consumption meters of the pilot area –one of each for the whole pilot area–, which we will call *Heat* and *Fan*. They were represented by two output variables in the simulation model. There were no corresponding physical sensors for these variables, but their values can be directly derived from the BEMS temperature and air flow measurements.

### D. EVALUATION METHODOLOGY

Following the International Performance Measurement and Verification Protocol (IPMVP), our evaluation methodology compared energy savings achieved by the OPG with respect to a base case in which it is not used. This process was carried out both in the simulation environment and in the real building:

- Evaluation in the simulation environment: We selected 3 days in the 2016-2017 period, respectively corresponding to a prototypical average (12-Jan-2016), cold (21-Jan-2016), and warm day (30-Jan-2017) of the winter season. The baseline was the real operation of the building for the same days. These data were collected at the beginning of the project. More details of this procedure are described in Section V.A.
- Evaluation in the real building: The OPG was activated in the building during a 30-day period in the late winter season, from April 19th to May 19th 2017. The reason of this choice is that we identified in the simulation environment that the OPG can achieve better results in the transitions between seasons –usually, the heating season in Sanomatalo ends in the second week of May. For the baseline, we built a regression model from historical data which estimates the energy consumption of the HVAC system without the OPG from the weather and the occupancy values, following the recommendations in [67].

With this model, we obtained a reliable approximation of the consumption that would have been measured if the system without the OPG had been used during the real test period. More details of this procedure are described in Section V.B.

We also studied comfort in terms of the indoor temperatures (*IAT*) and $CO_2$ concentration (*Con*) mentioned above, checking that the simulated and measured values were within the acceptable ranges.

## IV. PREDICTIVE CONTROL ALGORITHM

### A. CONTROL SYSTEM ARCHITECTURE

The Operational Plan Generation module is the core of the control system in Energy IN TIME. It encompasses three main stages:

1) Collection of forecasted building data: The OPG retrieves the weather forecast and the occupancy prediction for the operation period, usually the next day. Within our project, the weather forecast was obtained from the Weather Analytics API,[5] and occupancy predictions were obtained by using an agenda, which identified working days and average room occupancy per hour.
2) Generation, simulation and evaluation of candidate plans: The OPG runs several simulations to reproduce the expected building behavior, in terms of energy consumption and comfort, under different operation plans and according to the forecasted conditions retrieved in the previous stage. The best plan in terms of energy efficiency and comfort satisfaction is selected. We explain in Section IV.B how these candidates to best plan are generated and assessed.
3) Storage and execution of best plan: The best plan is stored in a database and made available to the setpoint writing component, which eventually sends the OP control instructions to the BEMS. This database also stores the context associated to each selected OP, i.e. the forecasted building data used by the OPG algorithm and the simulation results. This information is useful to explain the rationale of an OP to the building managers, who can revise and modify the setpoint values in real time as well.

### B. OPERATIONAL PLAN GENERATION

Creation of alternative plans is performed by an iterative algorithm based on a greedy heuristic and extended to balance diversity and local optimization. In this section, we explain the main steps of this stage: (1) identification of situations of interest for energy savings or comfort improvement; (2) generation of candidate plans to address these situations; (3) candidate plans simulation and selection of the best one.

To illustrate the processing in the OPG, we will assume a case with only one *Tsupply*setpoint, in which decrementing the setpoint means reducing the IAT and the energy consumption. We will also center the explanation in type A situations (see below). Nevertheless, the same principle applies to problems involving multiple variables and situations B and C. The explanation can be easily extended to more than one (independent) variable.

The overall functioning of the algorithm is depicted in Fig. 2, and its details are covered in the following subsections.

#### 1) IDENTIFICATION OF SITUATIONS OF INTEREST

In Fig. 3, we show an optimization scenario in which the IAT is controlled by a single *Tsupply* setpoint, as in our building. Given an initial plan for *Tsupply* values, we can simulate it and identify opportunities for energy optimization:

1) 8:00 – 12:00: Heating control results in an IAT above the upper comfort threshold. The previous *Tsupply* setpoints must be reduced to guarantee comfort.
2) 7:00 – 13:00: Heating control results in an IAT within the comfort range, but it may be possible to reduce the previous *Tsupply* setpoints while keeping the temperature above the lower bound of the comfort threshold.

Note that in both situations setpoint decrement may not be possible if it is already at the minimum value allowed by the equipment.

Analogously, we can identify one situation in which more energy is required, since the comfort requirements are not satisfied:

1) 15:00 – 18:00: Heating control results in an IAT below the upper comfort threshold. The *Tsupply* setpoint must be increased

Note that in this case setpoint increment may not be possible if it is already at the maximum value allowed by the equipment.

#### 2) GENERATION OF CANDIDATE PLANS

Let us consider a time instant $t$, and the corresponding setpoint value at this time $s_t$. For example, in Fig. 3, let us suppose $t = 9{:}00$; hence, $s_t = 23$ for *Tsupply* setpoint. We notate setpoint values at time $t - \Delta t$ as $s_{t-\Delta t}$; e.g. if $\Delta t = 4$, then $s_{t-4} = 20$, considering 1-hour intervals for simplicity's sake.

Let us notate the modification of a setpoint value $s$ as $\hat{s} = s \pm \Delta s$; e.g. decrementing $s_t$ in $\Delta s = 0.5$ give us $\hat{s}_t = 23 - 0.5 = 22.5$. The sets $\{\Delta t\}$ and $\{\Delta s\}$ are discrete and ordered. We define a time horizon $\Delta t^{max} = \max\{\Delta t\}$ to limit the temporal window of the modifications, as well as a maximum setpoint change value $\Delta s^{max} = max\{\Delta s\}$.

The candidate plan generation process starts from the current best plan, which at the beginning can be predefined or roughly estimated from outdoors temperatures. Next, it detects a situation of interest by analyzing the simulation of the current best plan; e.g. in our example, situation A at $t = 9{:}00$. For this $t$, the algorithm will propose a few candidate plans by decrementing previous setpoint values.

[5] http://dev.weatheranalytics.com

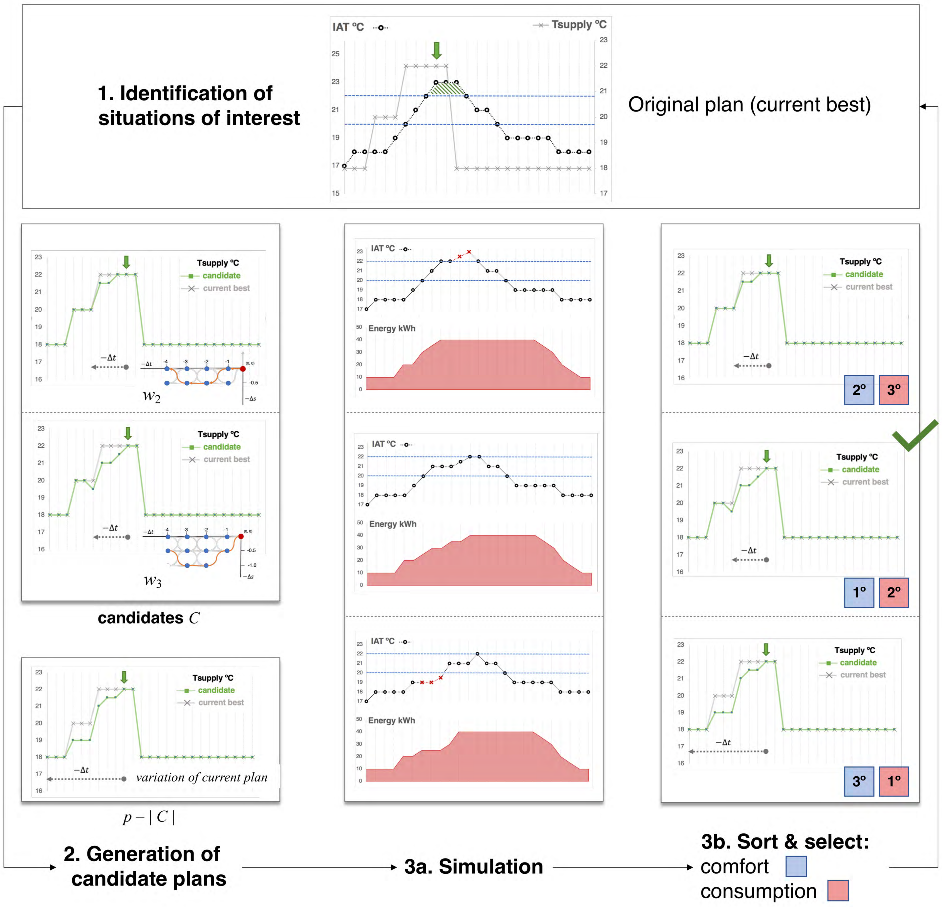


**FIGURE 2. Overall functioning of the OPG algorithm, including main stages: (1) identification of situations of interest (Section IV.B.1); (2) generation of candidate plans (Section IV.B.2); (3) plan simulation (IAT outside the comfort range during office hours is marked with ×) and assessment (Section IV.B.3). The second candidate plan is selected, because it has the best comfort ranking.**

To model all possible combinations of setpoint modifications in situation A, we define a lattice graph like the one in Fig. 4. Each vertex of this graph represents a setpoint modification at a given previous instant: $\hat{s}_{t-\Delta t} = s_{t-\Delta t} - \Delta s_{t-\Delta t}$. Each directed edge connects a setpoint change with the following setpoint change in reverse time order.

From this graph, the setpoint modifications that form a candidate plan are modeled as the result of a random walk[6] through this graph $w = \langle \hat{s}_t, \hat{s}_{t-1}, \hat{s}_{t-2}, \ldots, s_{t-\Delta t^{max}} \rangle$, with the following properties:

1) The walk starts at (0, 0) node, representing the current setpoint at starting time $t$ (i.e. the current setpoint is not modified)
2) Each step goes from $t'$ to $t' - 1$ for any $t'$ in the sequence (i.e. always moving from right to left in the graph)
3) The length of each path is $|\{\Delta t\}|$ (i.e. each path is a sequence of setpoint changes from $t$ to $t - \Delta t^{max}$)
4) The transition probability at each step from $\hat{s}_{t'}$ to $\hat{s}_{t'-1}$ is given by the following function (Eq. 1):

$$p(\hat{s}_{t'} \rightarrow \hat{s}_{t'-1}) = \begin{cases} \delta & \textit{if } \Delta s_{t'} = \Delta s_{t'-1} \\ \frac{1-\delta}{|\{\Delta s\}|-1} & \textit{otherwise} \end{cases} \tag{1}$$

with the diversification parameter $\delta \in [0, 1]$. This function balances two choices: maintaining the same previous setpoint change ($\delta$) and selecting any setpoint change ($1-\delta$). If $\delta = 0$, the transition probabilities at each step are the same for each allowed direction. If $\delta \gg 0$, the setpoints will tend to decrease in the same amount.

An identical graph is built in situation B. An analogous graph and a corresponding probability function are defined in situation C to represent setpoint increments.

In Fig. 5, we depict two examples of random walks and the resulting setpoint modification sequences $w_1, w_2$.

To reduce the number of possible alternatives, we can introduce an additional restriction to the walks:

1) Only moves to *closest* nodes in horizontal, vertical and diagonal directions are allowed (i.e. differences between time instants of changes of $\Delta s$, if any, are small)

[6] A random walk is a path consisting of a sequence of random steps on a mathematical space. Formally, it can be defined as a sum of a sequence of independent, identically distributed random variables representing move directions, or as a Markov chain over the subjacent state space [68].

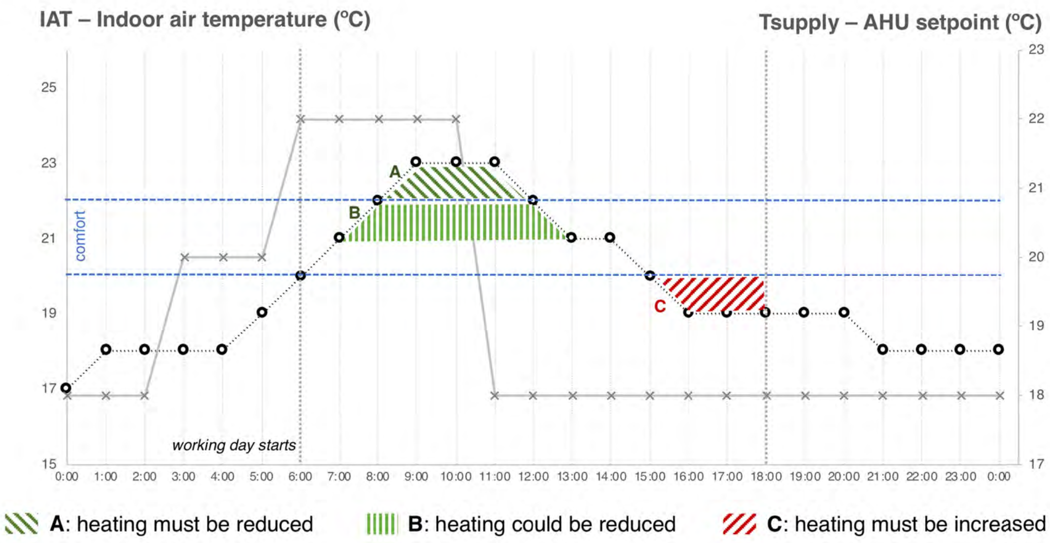


**FIGURE 3.** Identification of savings opportunities and discomfort in a simulated plan: indoor temperature ⋯○⋯ vs *Tsupply* setpoint values —×—. The comfort range in [20, 22] °C is also shown (dashed line).

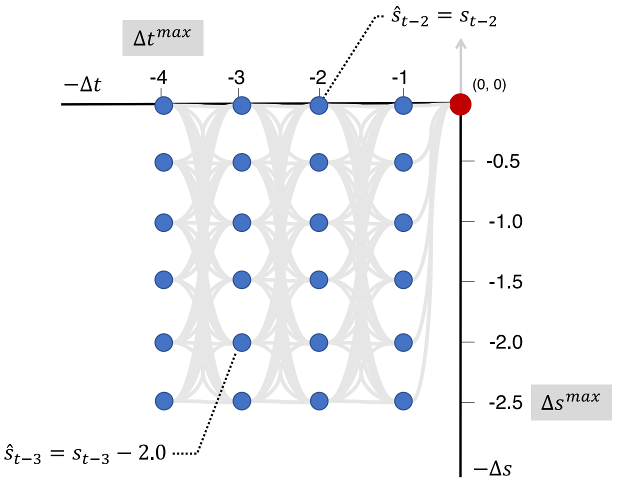


**FIGURE 4.** Lattice graph representing possible setpoint modifications (situation A, *Tsupply*). Time intervals are set to 1 hour, setpoint modifications are multiples of 0.5 °C.

This assumption considerably reduces the number of possible walks, as shown in Fig. 6. Note that this restriction may prevent the algorithm to explore the complete range of allowed $\{\Delta s\}$ modifications. Moreover, other heuristics could be incorporated to the process by means of additional walk restrictions encoded in the transition probability function; e.g. to limit how many different $\Delta s$ can be used in the same walk.

Our implementation of the OPG considers two particular situations: pre-conditioning and post-conditioning. Pre-conditioning is performed to achieve comfort at the beginning of the working day, while post-conditioning is performed to save energy by relaxing the comfort requirements at the end of the working day and later. We apply predefined setpoint change strategies for each variable during these intervals, which allow us to reduce the number of required simulations.

### 3) CANDIDATE PLANS SIMULATION AND SELECTION

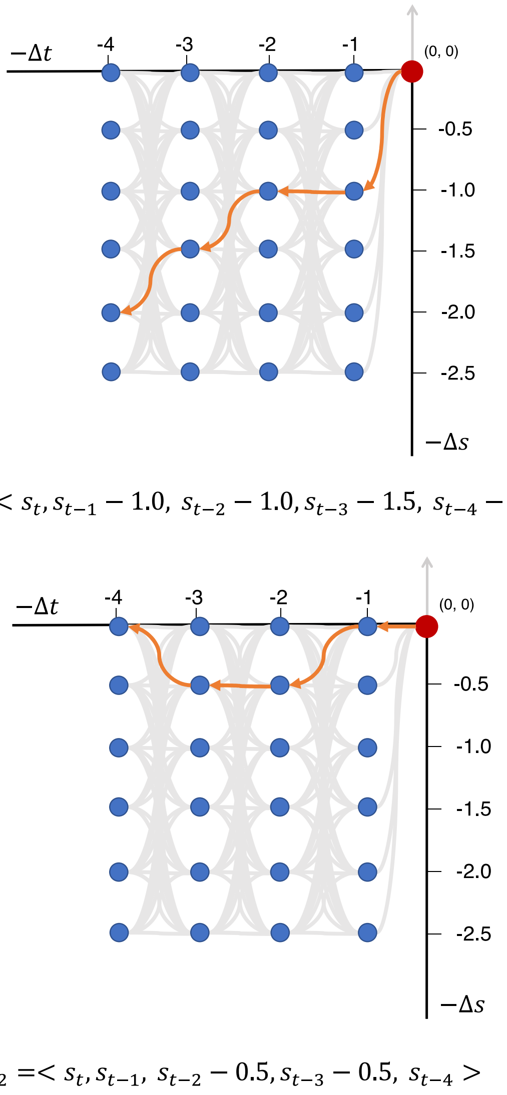
Each $w \in W$ produces a candidate OP, which is built by replacing the appropriate setpoints of the current best plan by $\left\{\hat{s}_{t-1}, \hat{s}_{t-2}, \ldots, \hat{s}_{t-\Delta t^{max}}\right\}$. The remaining setpoints outside the $[t - \Delta t^{max}, t-1]$ interval are not modified. The algorithm only selects a small random subset of candidate OPs $C \subseteq W$ to be simulated. In the best case, all the OPs in $C$ will be simulated in parallel; therefore, the selection of $C$ may depend on the simulator capabilities (see the experimental setup in Section V).

Finally, the algorithm picks the most efficient OP satisfying comfort requirements. Efficiency is calculated as the total energy consumption of the plan, while comfort can be measured in different ways; for example, by using the root-mean-square deviation (RMSD) or the % of time with comfort-related values (e.g. *IAT*, *Con*) inside the comfort range, maybe limited to a period of interest (e.g. office hours). If there is no such plan, the OPG selects the closest one to meet the requirements. To do so, OPs are firstly sorted by comfort satisfaction, and secondly by energy consumption.

The procedure is restarted to identify the next interesting situation (Section IV.B.1), now using the simulation of the new plan as a reference. The algorithm iterates while there are remaining situations to process or when a maximum number of situations have been processed.

### 4) TRIGGERING THE OPG ALGORITHM

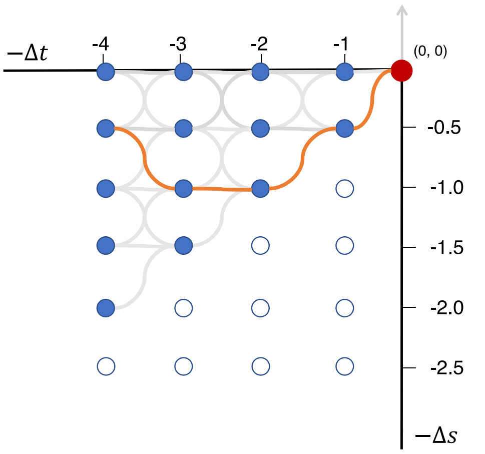
The OPG algorithm is usually launched before midnight to calculate the setpoints for the next day, allocating enough time to let the process finish before setpoints are due – a few hours in most cases. The algorithm can run again several times during the day, in order to create a new plan for the remainder of the day using updated weather and occupancy predictions and to recover from control deviations and failures.



$w_1 =< s_t, s_{t-1} - 1.0,\ s_{t-2} - 1.0, s_{t-3} - 1.5,\ s_{t-4} - 2.0 >$

$w_2 =< s_t, s_{t-1},\ s_{t-2} - 0.5, s_{t-3} - 0.5,\ s_{t-4} >$

**FIGURE 5. Samples of setpoint modification sequences obtained by using random walks, restrictions (A)–(D) apply.**



$w_3 = \langle s_t, s_{t-1} - 0.5,\ s_{t-2} - 1.0,\ s_{t-3} - 1.0,\ s_{t-4} - 0.5\rangle$

**FIGURE 6. Simplified setpoint modification graph and random walk sample, restrictions (A)-(E) apply.**

This formulation slightly diverges from the receding horizon control typically implemented in canonical MPC, because the prediction horizon is not shifted. However, note that such a receding horizon could be implemented just by generating control instructions for a whole period (e.g. 24 hours) each time the OPG algorithm is triggered, instead of generating control instructions until the end of the current day.

At the moment, the implemented recalculation process only generates a baseline plan using predefined operation curves when an updated weather forecast significantly differs from the initially used one. Enabling a faster and maybe simplified version of the OPG for quick recovery during the day, triggered by different events –e.g. comfort degradation is detected with live BEMS data–, remains as future work.

### C. COMPUTATIONAL PROPERTIES

The OPG algorithm cannot guarantee a global optimum in terms of energy consumption for two reasons: (a) only a limited number of setpoint modifications are explored; (b) the global plan is built from locally pseudo-optimal choices focused on situations of interest A, B, C. Conversely, it yields good solutions in a reasonable time, and allows easy incorporation of heuristics in the setpoint modification process.

Regarding (a), in the general formulation (Fig. 4 and 5), the number of possible setpoint change sequences $|W|$ for each situation and independent variable is bounded by $|W| \leq |\{\Delta s\} + 1|^{|\{\Delta t\}|}$. In the restricted formulation (Fig. 6), the number of possible random walks is bounded by $|W| < 3^{|\{\Delta t\}|}$. The $|W|$ for multiple-dimension random walks grows exponentially [68]. In any case, only $|C| \ll |W|$ candidate OPs will be simulated at each iteration. Therefore, the overall efficiency of the algorithm is bounded by the number of situations of interest processed multiplied by the time required to run each batch of simulations of size $|C|$. Note that the execution time of the OP generation process is insignificant compared to the simulation time.

The parallel cloud version of IESVE allows running a fixed number $p$ of parallel simulations without performance degradation. To increase solution diversity, we can set $|C| < p$, and our implementation will fill the remaining simulation slots with other plans, namely: (a) random variations of the current best OP at any time before $t$; (b) combinations of previously discarded good OPs; (c) baseline OPs –e.g. for Sanomatalo, OPs based on outdoors temperature. These plans are compared against the plans obtained with the random walks, and can be selected as best current plan for the next iteration in the same conditions.

Regarding (b), under some realistic assumptions, the OPG algorithm finds a good approximation to the optimal solution. Specifically, for *Tsupply* control in Sanomatalo we can assume that external temperatures and internal occupancy values follow a bell-shaped curve. To guarantee comfort in the winter season, the optimal OP would entail increasing the temperature setpoints in the early morning, then decreasing them around noon, and maybe incrementing them again in the afternoon. Moreover, the low external temperatures favor heat losses, which would in turn require supplying hot air frequently. Therefore, it is safe to limit the search to local setpoint increments and decrements.

## V. EXPERIMENTS AND RESULTS

The OPG algorithm has been implemented in the Python and R programming languages. For the experiments in this section, it ran on a Supermicro SuperServer 6027R-TRF, configured with 2 processors Intel Xeon E-2600 2.4GHz (2 × 8 cores), 128 GB RAM, 2 × 600 GB magnetic storage. The details of the cloud-based version of the IESVE simulator are not disclosed by IES by confidentiality reasons.

Experimentation based only in the simulation environment was performed in advance to test and tune the deployment of the OPG used in the real building. After some preliminary tests and following the building requirements, the OPG parameters were set to the following values:

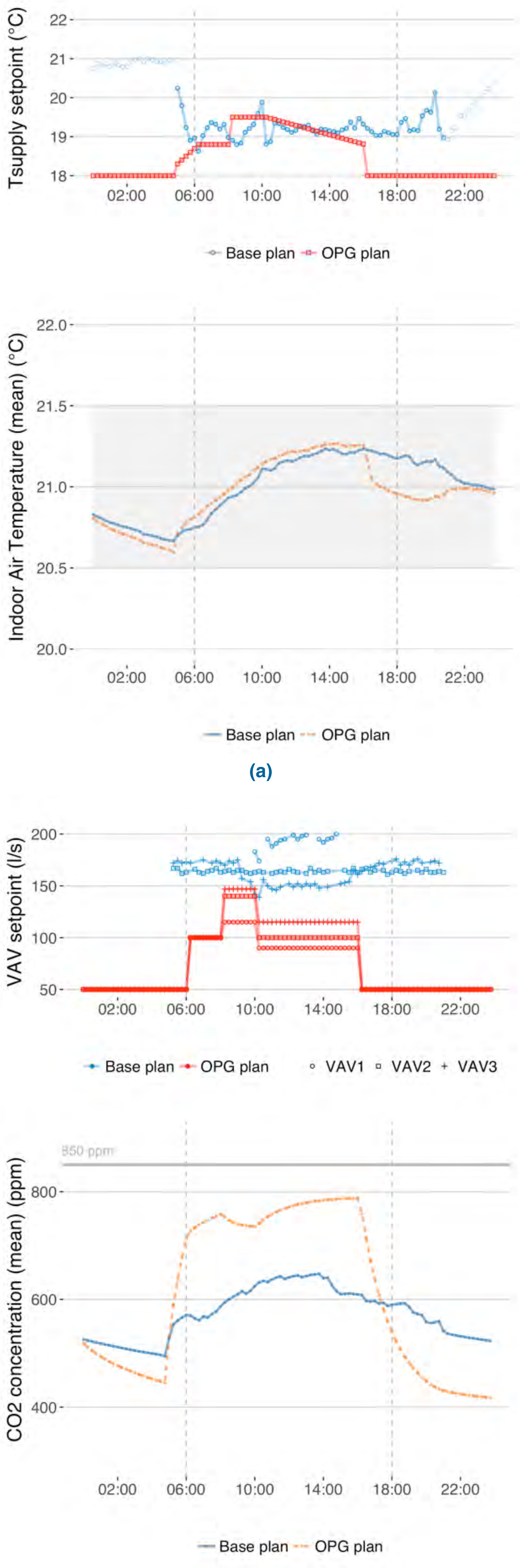
- Simulation batch size: $p = 50$, resulting in simulation times below 20 minutes
- Ahead period of the OPG: 1 day, no receding horizon
- OPG starting time: $>$ 2 hours before the first setpoint is due
- Maximum setpoint change frequency: 15 minutes
- Simulation output resolution: 15 minutes
- Only minor setpoint changes are allowed in random walks
- $\{\Delta t\}Tsupply, VAVairflow = \{30, 60, 90, 120\}$ minutes
- $\{\Delta s\}Tsupply = \{0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$ °C
- $\{\Delta s\}VAVairflow = \{0, 25, 50\}$ l/s
- $\delta = 0.3$ (random setpoint modifications are preferred)
- Tsupply and VAV setpoints are optimized independently (first *Tsupply* and then *VAVairflow*)
- Comfort satisfaction is measured by using the RSMD from the comfort interval

Additionally, the maximum number of simulation batches was restricted in order to establish an upper limit to the execution time. Since an average simulation batch took 20 minutes (with $p = 50$), we set the maximum number of batches per plan to 6 in order to keep the execution time under 2 hours. This means that 6 A-B-C situations (Figure 3) can be analysed in each run of the OPG. Excluding pre- and post-conditioning, 4 out of 6 were reserved for *Tsupply* changes, and 2 for *VAVairflow* changes. Situations are sorted by relevance at each iteration of the OPG algorithm; e.g. for *Tsupply*, situations A and C are more important than B.

### A. SIMULATION ENVIRONMENT

As described in Section II.D, we selected three prototypical days of the winter season: average (Standard), cold (Harsh) and warm (Intermediate). Then, we simulated the behaviour of the pilot area of the building according to the setpoints originally applied (i.e. the base plan) and the setpoints calculated by our algorithm (i.e. the OPG plan), in order to check how they compare in terms of comfort and consumption.

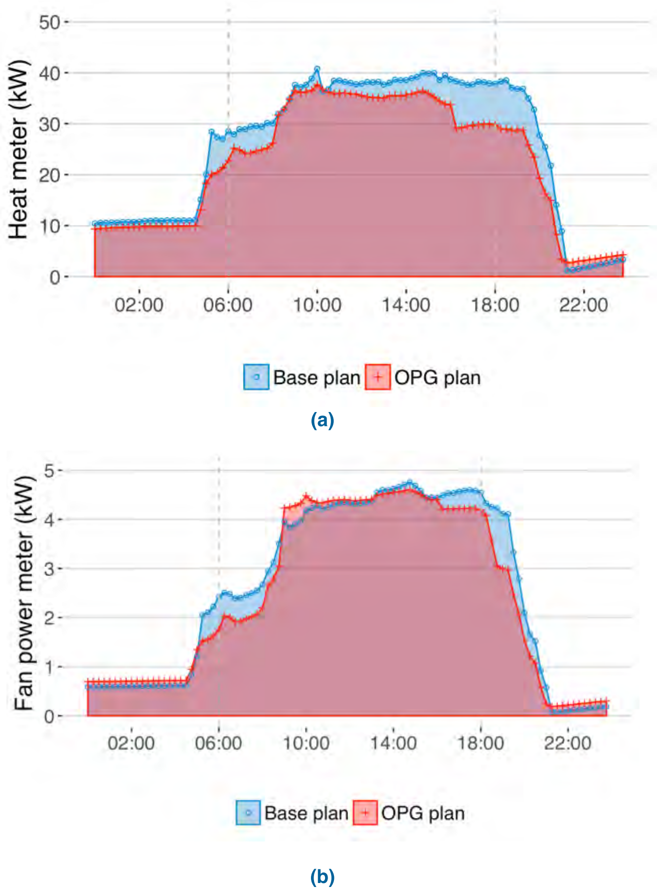
Fig. 7 depicts the simulation results for a Standard day, corresponding to the most common conditions during the winter season. In the top of the figure, we show the setpoints of the base and the OPG plans for the *Tsupply* and the $VAVairflow_i$ operation –the main working hours are delimited. In both cases, the largest operation differences correspond to



(a)

(b)

**FIGURE 7. Comparison of baseline and OPG plans in terms of setpoints and comfort values (with comfort thresholds) for a Standard winter day, simulation environment: (a) *Tsupply* operation; (b) $VAVairflow_i$ operation.**



(a)

(b)

**FIGURE 8. Comparison of power (kW) of baseline and OPG plans for a STANDARD winter day, simulated environment: (a) heat meter; (b) VAV fan power meter.**

the less crowded periods. Note that the *Tsupply* setpoints of the base plan before 5:00 and after 21:00 are registered but not applied; control is managed by a human-operated switch. In the bottom of the figure, we show the mean comfort values obtained in simulation in terms of *IAT* and *Con*; these values lay within the comfort intervals (also included in the figure).

Fig. 8 shows the power consumption calculated by the simulation model; respectively, the *Heat* and *Fan* power meters values. We can observe that most savings are achieved at the borderline hours, that is, at the beginning and at the end of the working day. This is consistent with the pre-conditioning and post-conditioning provisions made by the OPG algorithm.

**TABLE 1. Energy consumption (kWh) for the experiments in the simulation environment.**

| | STANDARD<br>12-Jan-2016 | | HARSH<br>21-Jan-2016 | | INTERMEDIATE<br>30-Jan-2017 | |
|---|---|---|---|---|---|---|
| | Heat | Fan | Heat | Fan | Heat | Fan |
| Base | 618.66 | 62.47 | 896.64 | 79.88 | 207.95 | 30.99 |
| OPG | 539.77 | 59.23 | 854.03 | 74.41 | 132.98 | 32.46 |
| Savings (%) | **12.75** | **5.18** | **4.75** | **6.85** | **36.05** | **-4.75** |

Table 1 includes the detailed numbers for the three reference days. To obtain the overall energy consumption, we have approximated the integral of the power functions with the area under the curve (AUC). AUC has been computed by: (1) interpolation of data points with a spline; and (2) calculation of the adaptive quadrature of the interpolated function [69].

It can be seen that the OPG reduces the power consumption of the base operation of both the heating and the ventilation subsystems. As expected, in the experiments the highest heating savings are achieved in the warmer intermediate day, when there is still room for adjustments. Broadly speaking, the OPG dynamically adapts the operation to the particular conditions of a specific day without requiring the operator attention, which is convenient in less cold days in the winter season or before transitioning to the spring season. Conversely, the HVAC system is already operating at (almost) full power during the harsh days to achieve comfort, and therefore there is little room for improvement during working hours. A more detailed discussion on these features is included in Section VI. On the other hand, fan power savings have similar values in different working days. The bad results in the intermediate day were the consequence of the misestimation of the occupancy used by the algorithm.

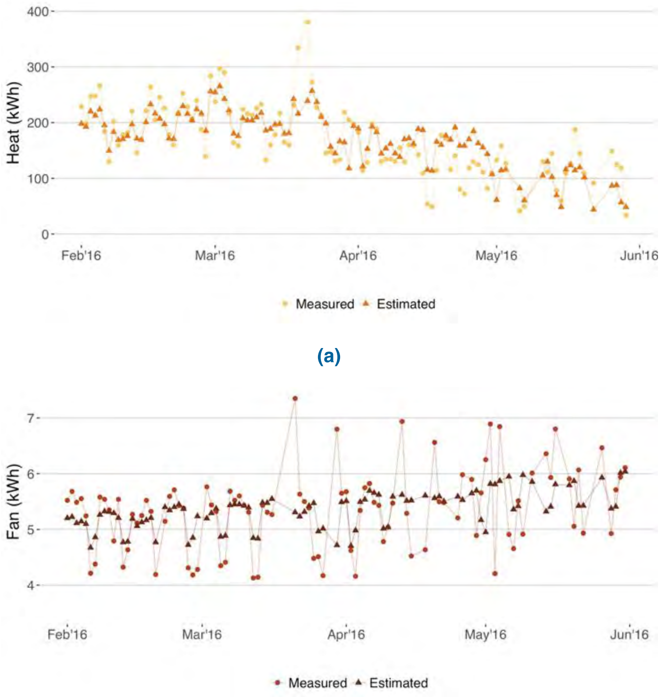
### B. ON-SITE TEST AND EVALUATION

The evaluation in the pilot area of the real building was performed from April 19th to May 19th 2017. These days mostly fit into the Intermediate category studied in the previous section, the one which yielded the highest energy savings.

The baseline for daily energy consumption was calculated by a generalized linear regression model (glmnet) [70], a method based on lasso analysis (least absolute shrinkage and selection operator). Other prediction techniques, such as linear regression or autoregression, could have also been explored. Source data for the model was obtained from building sensors (energy, OAT and occupancy) logged in the period February-May 2016.

More specifically, we developed two baseline models for prediction of daily consumption of heating equipment and VAV fans, based on the expected heating demand and occupancy. Expected daily energy demand (*hdd*, in heating degree days) was calculated by using integration with base temperature set to 18 °C and the BEMS OAT [71]. Estimated daily occupancy (*occ*, in %) was the maximum occupancy value of the office agenda. To build the prediction models, we firstly pre-processed the data, discarding outliers and measurement errors.

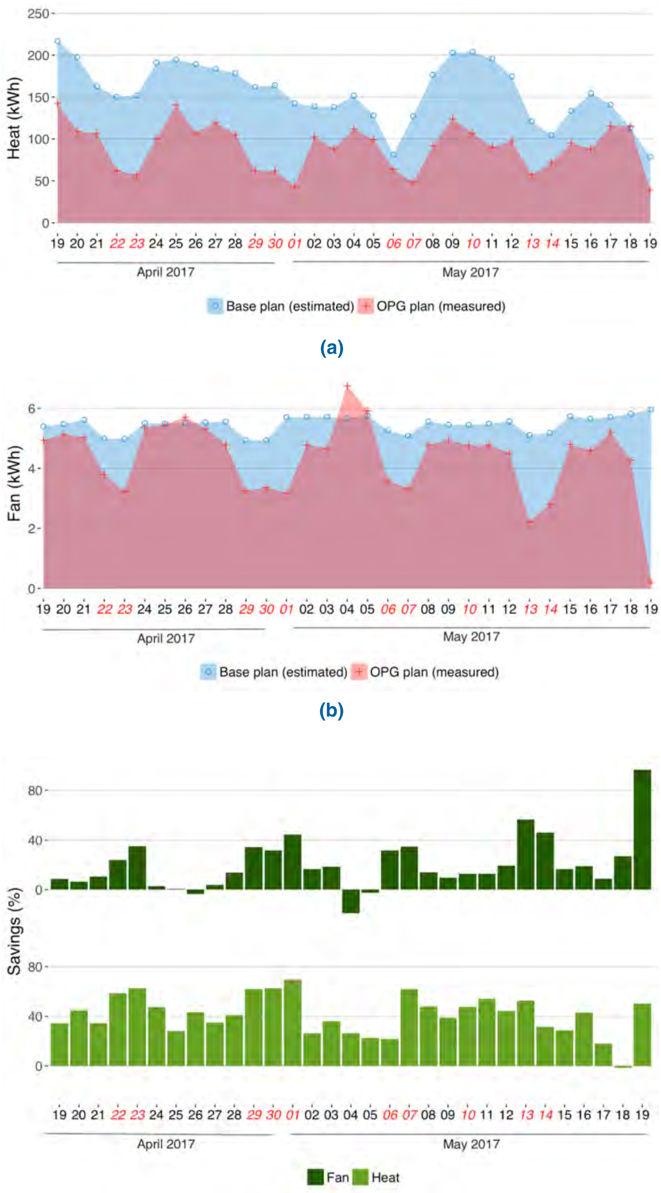
Fig. 9 compares the energy consumption in February-May 2016 and the values calculated by the heating and the fan consumption prediction models. The parameters of the regression models are given in Eq. 2 and Eq. 3 respectively, yielding correlation coefficient values of $R^2 = 0.632$ and $R^2 = 0.234$. Note that: (1) the heating baseline model slightly overestimates consumption from mid-April to June, which means that energy savings calculated in the next section are slightly overestimated as well; (2) the fan power model has



(a)

(b)

**FIGURE 9. Comparison of daily energy consumption (kWh) estimated by the baseline models vs historical data in Feb-May 2016: (a) heating; (b) VAV fans.**



(a)

(b)

(c)

**FIGURE 10. Comparison of daily energy consumption (kWh) during the test period vs estimated by the baseline models: (a) heating; (b) VAV fans; (c) savings. Days in red italic font are weekend or holiday days.**

a low $R^2$ value, which means that energy savings calculated with this model should be considered with caution.

$$Heat^{*}\,(hdd\,,\,occ) = -9.122 \times hdd + 0.703 \times occ + 20.71 \quad (2)$$

$$Fan^{*}\,(hdd\,,\,occ) = -0.038 \times hdd + 0.012 \times occ + 5.015 \quad (3)$$

Fig. 10(a) and 10(b) show the comparison of the values of daily energy consumption in the pilot area obtained from the BEMS (+) with the values estimated by the prediction models (o) for the test period in the real building. Fig. 10(c) shows the energy savings achieved in % of the (estimated) consumption before optimization.

**TABLE 2. Energy savings (kWh) achieved in the on-site test with the OPG control vs estimated by the baseline models.**

| | Overall | | Workdays | | Weekends & Holidays | |
|---|---|---|---|---|---|---|
| | Heat | Fan | Heat | Fan | Heat | Fan |
| Estimated (daily avg) | 156.47 | 5.47 | 164.70 | 5.62 | 132.79 | 5.06 |
| OPG (daily avg) | 91.12 | 4.37 | 101.77 | 4.78 | 60.50 | 3.20 |
| Savings (daily avg %) | **41.76** | **20.12** | **38.21** | **14.91** | **54.44** | **36.73** |

As summarized in Table 2, the average savings per day are, respectively, around 40% for the thermal subsystem and around 20% for the electrical subsystem. Weekends and holidays offer opportunities for higher energy savings, since the OPG adjust the operation to the building occupancy better than the manual operation.

Savings have been achieved without compromising users' comfort. Figure 11 shows the *IAT* and $CO_2$ concentration values in the pilot area in the evaluation period. The *IAT* values were calculated as follows: (1) sensor measurements, obtained from the BEMS temperature sensors (25), were resampled and interpolated to match the setpoint change frequency parameter (15 minutes); (2) sensor temperatures were averaged at each timestamp; (3) maximum and minimum values of timestamps within the working hours were

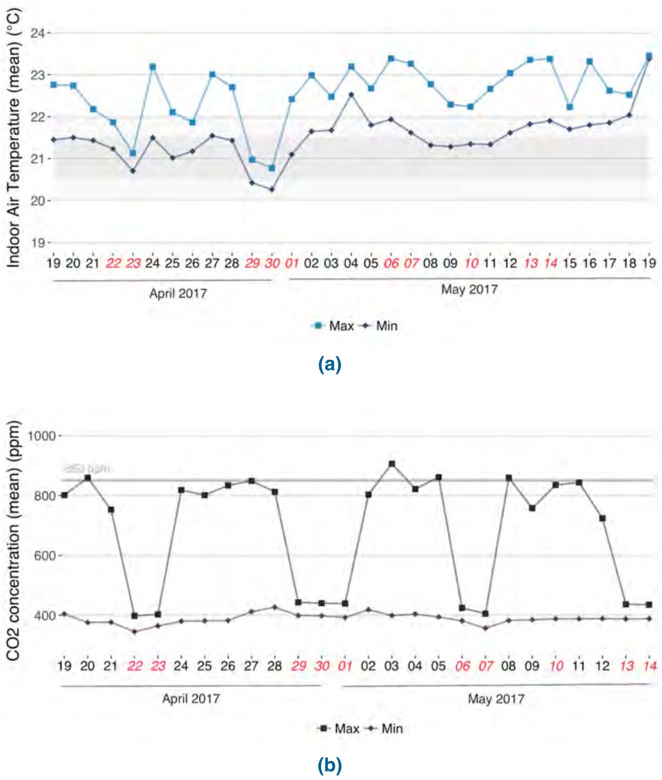


**FIGURE 11. Daily comfort values achieved in the on-site test with the OPG control, maximum and minimum sensor average values: (a) indoor air temperature (*IAT*, °C), with comfort interval; (b) $CO_2$ concentration (*Con*, ppm), with comfort threshold. $CO_2$ measurements were only available from April 19th to 14th May. Days in italic red font are weekend or holiday days.**

obtained. *Con* values were retrieved from 4 offline sensors; the remainder of the procedure is the same as for *IAT*.

*IAT* min values lie within the comfort range during the test period. Actually, it would have been possible to configure the OPG to reduce *Tsupply* even more. However, as explained at the beginning of this section, we prioritized optimizing discomfort situations. *IAT* max values are over the comfort upper threshold by 1 °C. A more detailed analysis of these values identified that discomfort was not sustained and only happened for short time periods (less than 1 hour).

Similarly, $CO_2$ concentration values are mostly below the comfort threshold, although with some exceptions. After more detailed analysis, we identified that the highest values were measured by a single sensor, which in some cases exceeded 950 ppm. However, the levels calculated by the simulation environment were considerably smaller. This is a clear example of the importance of having all the sensor data available through the BEMS. With the $CO_2$ sensors offline, our system was not able to recalibrate the simulation model –which would have led to better plans–, nor to detect in real time that some plans were not guaranteeing comfort –which would have triggered a correction action.

## VI. DISCUSSION

The results presented in Section V show that the use of MPC in the offices of the Sanomatalo building significantly reduces energy consumption; particularly heating consumption. Automatic control allows for more effective plans since it enables a finer-grained and more frequent scheduling of setpoint changes without the supervision of the building operators. The OPG algorithm and software offer a flexible and configurable framework to generate more efficient operation plans, predicting the building state and adapting energy usage to more realistic demand estimations without compromising users' comfort. As expected, it has proved to be particularly successful in optimizing temperature setpoints, in which a longer control horizon, accounting for the inertia of the equipment, is crucial. The building operators were satisfied by the use of the system during the test period. One highlighted system's feature was the capabilities to validate the plans in advance (and even to modify them) and to provide justifications of the algorithm decisions –by means of graphical depictions of the simulation results, in a similar way to Fig. 11.

As already anticipated by the experiments in the simulation environment (Section V.A), the highest energy savings can be obtained for heating in the mid-season, when it is not necessary to use the heating equipment at full, and particularly, in the warmer days (Table 1, Intermediate). At the same time, the system can react to isolated cold days. The on-site evaluation in the Sanomatalo building, which was carried out at the end of the 2017 heating season, confirmed these assumptions. Energy usage in colder days could be even more optimized by relaxing the comfort temperature restrictions to permit OPs with some minor discomfort for a short period of time. The advantage of our system is that it allows operators to characterize and quantify this discomfort in advance, thus supporting them to make more informed decisions. (Note that this feature was not exploited in the experiments.)

Our system reduced the temperature setpoints given by the normal operation of the building between 0.5 and 2 °C. During the on-site test, this meant savings in heating above 40% (Table 2) while keeping comfort (see Fig. 11(a)). The algorithm adapted well to workdays and weekends, showing slightly better results in the former ones (Fig. 10(c)). A possible explanation for this is that operators have lower availability in weekends and holiday days, and therefore it is not possible for them to create customized plans. The airflow consumption was also reduced in a 20% (Table 2) without compromising the $CO_2$ concentration comfort (Fig. 11(b)), despite the lack of a proper model calibration and the smaller number of simulations involving VAVs. Nevertheless, due to the lower accuracy of the baseline model, these results are less precise and should be further analyzed; e.g. by using autoregression to build the baseline [72].

In summary, although the Sanomatalo building was already efficiently operated, and considering the limitations of the baseline estimations, the overall savings figures in the intermediate winter are in line with the 30% target of EU energy directive [5] and the 35% savings estimations provided in [7]. The experiments also revealed more opportunities for savings in the future, e.g. by improving the simulation model with online $CO_2$ data (for calibration) and more detailed occupancy predictions (actual agenda data were not very fine-grained).

**TABLE 3. Energy savings (heating, MWh) in the pilot area projected for the whole year.**

| | STANDARD | HARSH | INTERM. | |
|---|---|---|---|---|
| Savings (%) | 12.75 | 4.75 | 36.05 | |
| Monthly consumption (MWh) | 7 | 9 | 4 | |
| # months per year | 3 | 2 | 3 | **TOTAL** |
| Reduction (MWh) | 2.6775 | 0.855 | 4.326 | **7.8585** |

In Table 3, we show a rough projection of energy savings in the pilot area for the whole year using: (1) savings calculated in Section V.A (only heating); (2) historical monthly consumption values provided by the building operators; (3) estimated distribution of day types. We assume that the heating system is not used during the summer, and therefore it does not make sense to quantify savings in this period. It can be seen that the overall energy consumption reduction during the whole winter season is around 20%, larger than the consumption of a standard winter month. We can also estimate savings of $CO_2$ emissions: assuming a carbon factor of 206 $kgCO_2$/MWh for district heating energy in Finland [73], the new system applied in the pilot area can save more than 1.60 Tons of $CO_2$ per year. These figures could be directly adapted to other estimation of day types (e.g. including savings in the summer) and extended to other sections of the building with similar configuration.

The implementation of the system in other buildings entails: (a) developing a specific simulation model, if not available; (b) parametrizing the OPG algorithm, including the definition of energy optimization strategies; (c) finding appropriate sources for weather and occupancy forecasts; (d) adapting the setpoint writing component, if fully-automatic control is enabled; (e) deploying the computational infrastructure to run these components. As a matter of fact, in the context of the Energy IN TIME project we applied modified versions of the OPG to other scenarios, such as an airport and a hotel – achieving similar results, as briefly described in [66]. Among the tasks required to extend the system to other buildings, developing and tuning the simulation model is the most time-consuming one.

The collaboration with the Sanomatalo building operators revealed some prospective improvements to the system. First, it would be convenient to offer a better interface for configuration of the OPG and interaction with the generated plans, as in [74]. Second, users' comfort should be measured beyond thermal and $CO_2$ concentration intervals, probably by using PMV, adaptive comfort models and comfort standards [75], [76]. Third, occupancy estimations in our experiments were mostly static, while occupancy monitoring and reconfiguration have shown effective in the past [77], [78]. In this regard, the capabilities and limitations for OP recalculation during the day should also be further explored. Four, a more comprehensive study of energy savings with additional baseline models should be carried out in order to quantify more precisely the return of investing in our solution [79], in particular if only ventilation is addressed. Last but not least, the building setup was relatively simple, with district heating and almost fixed energy costs. It would be interesting to study the applicability and the scalability of the OPG approach to smart grids, including more control variables –some of them affecting the production side– and energy storage equipment; see for example [80], [81].

## VII. CONCLUSIONS AND FUTURE WORK

This paper has presented the design and the implementation of an MPC-based control system aimed at reducing energy consumption in non-residential buildings while guaranteeing occupants' comfort. The main difference of our proposal with respect to other approaches is that we use a full-complexity simulation model, which runs in parallel in the cloud. This allows using more accurate models and facilitates communication between computer scientists, building operators and simulation developers, exploiting synergies of their joint work. Comprehensive quantitative and qualitative comparison with MPC approaches using reduced-complexity simulation models would be useful to support decision-making between different alternative approaches.

Experimentation in the Sanomatalo building, located in Helsinki, both in the simulation environment and in the real building, has shown that important energy savings –up to 40% at the end of the winter season– can be achieved, particularly by optimizing the control of the heating equipment. Note that our approach can be adapted to other scenarios, and specifically, to cooling equipment. In our experiments we did not consider the energy costs of running our system, which should be deducted from the HVAC savings [82]. These promising figures can give rise to disruptive models for energy service provision, as we explore in [83].

The OPG algorithm opens several opportunities for further research. The current design relies on a variant of heuristic search, which can be hard to scale up if several variables are to be optimized at the same time. In this regard, other search and optimization techniques could be applied. Specifically, genetic algorithms allow balancing diversification and intensification of solution search by adjusting their parameters. Another possible extension of the OPG would be to incorporate means to define imprecise comfort ranges, thus formalizing the notion of relaxed comfort into the procedure. It would also be interesting to study how to represent energy optimization strategies in a machine-processable language, in such a way that the system could use them for self-configuration. Moreover, self-configuration could be supported by machine learning techniques able to identify successful operation patterns from historical data, and to apply reinforcement learning to reward and reuse particularly efficient OPG plans.

Finally, we believe that combining interpretable white/grey-box models, like the one used in this work, and efficient black-box models, learnt from historical data, is one of the most prospective directions for future work. Faster simulation of such hybrid model would allow for the implementation of more sophisticated optimization and planning techniques. Recent approaches to data-driven black-box models have showed good accuracy, but only for short time periods [84]. Learning more general and precise models would require larger datasets, more computational power, and techniques able to exploit them. Recent advances in the Deep Learning area suggest that this is a feasible goal.

## NOMENCLATURE

| | |
|---|---|
| **AHU** | Air Handling Unit |
| **AUC** | Area under the curve |
| **BCVTB** | Building Controls Virtual Test Bed |
| **BEMS** | Building Energy Management System |
| **C** | Set of candidate plans considered in an iteration of the OPG |
| **Con** | $CO_2$ concentration (parts per million, ppm) |
| $\delta$ | diversification parameter |
| $\Delta t$ | Time increment / decrement |
| $\{\Delta t\}$ | Set of time increment / decrement values |
| $\Delta t^{\mathbf{max}}$ | Maximum time increment / decrement |
| $\Delta s$ | Setpoint increment / decrement |
| $\{\Delta s\}$ | Set of setpoint increment / decrement values |
| $\Delta s^{\mathbf{max}}$ | Maximum setpoint value increment / decrement |
| **Fan** | Energy consumption due to electrical subsystem (kWh) |
| **Fan**$*$ | Estimated daily fan energy consumption with the baseline model (kWh) |
| **HDD** | Heating Degree Days |
| **hdd** | Estimated daily demand measured in HDD (integrated) |
| **Heat** | Energy consumption due to thermal subsystem (kWh) |
| **Heat**$*$ | Estimated daily heating energy consumption with the baseline model (kWh) |
| **HVAC** | Heating, Ventilation, and Air Conditioning |
| **IAT** | Indoor Air Temperature (°C) |
| **IPMVP** | International Performance Measurement and Verification Protocol |
| **MILP** | Mixed Integer Linear Programming |
| **MPC** | Model Predictive Control |
| **OAT** | Outdoor Air Temperature (°C) |
| **occ** | Estimated daily occupancy (maximum) (%) |
| **PMV** | Predicted mean value |
| **RSMD** | Root mean square deviation |
| $\mathbf{s_t}$ | Setpoint value at time t |
| $\mathbf{\hat{s}_t}$ | Setpoint value at time t modified |
| **t** | Time instant |
| **Tsupply** | AHU supply temperature setpoint value (°C) |
| **VAV** | Variable air volume unit |
| $\mathbf{VAVairflow_i}$ | Airflow setpoint value for VAV number *i* (liters per second, l/s) |
| **w** | Ordered sequence of setpoint changes from t to t $-\Delta t^{\max}$ |
| **W** | Set of all w |

## ACKNOWLEDGEMENTS

The authors would like to thank Vicente Madero (Acciona Construcción S.A.); Catherine Conaghan, Adalberto Guerra, Stephen Earle (Integrated Environmental Solutions Ltd.); Christian Beder (Cork Institute of Technology); and Jukka Heino (Caverion Suomi Oy) for their support and assistance with this research work.

## REFERENCES

[1] J. Laustsen, "Policy pathways: Energy performance certification of buildings," International Energy Agency (IEA), Paris, France, Tech. Rep., 2010.

[2] D. Urge-Vorsatz, K. Petrichenko, M. Staniec, and J. Eom, "Energy use in buildings in a long-term perspective," *Current Opinion Environ. Sustainability*, vol. 5, no. 2, pp. 141–151, 2013.

[3] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy Buildings*, vol. 40, no. 3, pp. 394–398, 2008.

[4] X. Cao, X. Dai, and J. Liu, "Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade," *Energy Buildings*, vol. 128, pp. 198–213, Sep. 2016.

[5] *Proposal for a Directive of the European Parliament and of the Council Amending Directive 2010/31/EU on the Energy Performance of Buildings*, Eur. Commission, Brussels, Belgium, 2016.

[6] *A Clean Planet for all—A European Long-Term Strategic Vision for a Prosperous, Modern, Competitive and Climate Neutral Economy*, Eur. Commission, Brussels, Belgium, 2018.

[7] A. Ghahramani, K. Zhang, K. Dutta, Z. Yang, and B. Becerik-Gerber, "Energy savings from temperature setpoints and deadband: Quantifying the influence of building and system properties on savings," *Appl. Energy*, vol. 165, pp. 930–942, Mar. 2016.

[8] V. Marinakis, H. Doukas, C. Karakosta, and J. Psarras, "An integrated system for buildings' energy-efficient automation: Application in the tertiary sector," *Appl. Energy*, vol. 101, pp. 6–14, Jan. 2013.

[9] A. Costa, M. M. Keane, J. I. Torrens, and E. Corry, "Building operation and energy performance: Monitoring, analysis and optimisation toolkit," *Appl. Energy*, vol. 101, pp. 310–316, Jan. 2013.

[10] B. Swords, E. Coyle, and B. Norton, "An enterprise energy-information system," *Appl. Energy*, vol. 85, no. 1, pp. 61–69, 2008.

[11] Y. Lu, S. Wang, and K. Shan, "Design optimization and optimal control of grid-connected and standalone nearly/net zero energy buildings," *Appl. Energy*, vol. 155, pp. 463–477, Oct. 2015.

[12] A. Afram and F. Janabi-Sharifi, "Theory and applications of HVAC control systems—A review of model predictive control (MPC)," *Building Environ.*, vol. 72, pp. 343–355, Feb. 2014.

[13] A. Mirakhorli and B. Dong, "Occupancy behavior based model predictive control for building indoor climate—A critical review," *Energy Buildings*, vol. 129, pp. 499–513, Oct. 2016.

[14] G. Serale, M. Fiorentini, A. Capozzoli, D. Bernardini, and A. Bemporad, "Model Predictive Control (MPC) for enhancing building and HVAC system energy efficiency: Problem formulation, applications and opportunities," *Energies*, vol. 11, no. 3, p. 631, 2018.

[15] D. B. Crawley, J. W. Hand, M. Kummer, and B. T. Griffith, "Contrasting the capabilities of building energy performance simulation programs," *Building Environ.*, vol. 43, no. 4, pp. 661–673, Apr. 2008.

[16] P. Rockett and E. A. Hathway, "Model-predictive control for non-domestic buildings: A critical review and prospects," *Build. Res. Inf.*, vol. 45, no. 5, pp. 556–571, 2017.

[17] S. Meyers and S. Kromer, "Measurement and verification strategies for energy savings certificates: Meeting the challenges of an uncertain world," *Energy Efficiency*, vol. 1, no. 4, pp. 313–321, 2008.

[18] A. Mahdavi, "Simulation-based control of building systems operation," *Building Environ.*, vol. 36, no. 6, pp. 789–796, 2001.

[19] J. A. Clarke *et al.*, "Simulation-assisted control in building energy management systems," *Energy Buildings*, vol. 34, no. 9, pp. 933–940, 2002.

[20] S. Petersen and S. Svendsen, "Method for simulating predictive control of building systems operation in the early stages of building design," *Appl. Energy*, vol. 88, no. 12, pp. 4597–4606, 2011.

[21] M. Killian and M. Kozek, "Ten questions concerning model predictive control for energy efficient buildings," *Building Environ.*, vol. 105, pp. 403–412, Aug. 2016.

[22] W. Z. Huang, M. Zaheeruddin, and S. H. Cho, "Dynamic simulation of energy management control functions for HVAC systems in buildings," *Energy Convers. Manag.*, vol. 47, nos. 7–8, pp. 926–943, 2006.

[23] R. Z. Freire, G. H. C. Oliveira, and N. Mendes, "Predictive controllers for thermal comfort optimization and energy savings," *Energy Buildings*, vol. 40, no. 7, pp. 1353–1365, 2008.

[24] D. Agdas and R. S. Srinivasan, "Building energy simulation and parallel computing: Opportunities and challenges," in *Proc. Winter Simulation Conf.*, 2014, pp. 3167–3175.

[25] G. Bianchini, M. Casini, A. Vicino, and D. Zarrilli, "Demand-response in building heating systems: A model predictive control approach," *Appl. Energy*, vol. 168, pp. 159–170, Apr. 2016.

[26] J. Figueiredo and J. S. da Costa, "A SCADA system for energy management in intelligent buildings," *Energy Buildings*, vol. 49, pp. 85–98, Jun. 2012.

[27] J. Široký, F. Oldewurtel, J. Cigler, and S. Prívara, "Experimental analysis of model predictive control for an energy efficient building heating system," *Appl. Energy*, vol. 88, no. 9, pp. 3079–3087, 2011.

[28] H. Huang, L. Chen, and E. Hu, "A new model predictive control scheme for energy and cost savings in commercial buildings: An airport terminal building case study," *Building Environ.*, vol. 89, pp. 203–216, Jul. 2015.

[29] J. Ma, J. Qin, T. Salsbury, and P. Xu, "Demand reduction in building energy systems based on economic model predictive control," *Chem. Eng. Sci.*, vol. 67, no. 1, pp. 92–100, 2012.

[30] S. J. Kang, J. Park, K.-Y. Oh, J. G. Noh, and H. Park, "Scheduling-based real time energy flow control strategy for building energy management system," *Energy Buildings*, vol. 75, pp. 239–248, Jun. 2014.

[31] I. Hazyuk, C. Ghiaus, and D. Penhouet, "Optimal temperature control of intermittently heated buildings using model predictive control: Part II—Control algorithm," *Buildings Environ.*, vol. 51, pp. 388–394, May 2012.

[32] R. De Coninck and L. Helsen, "Practical implementation and evaluation of model predictive control for an office building in Brussels," *Energy Buildings*, vol. 111, pp. 290–298, Jan. 2016.

[33] S. C. Bengea, A. D. Kelman, F. Borrelli, R. Taylor, and S. Narayanan, "Implementation of model predictive control for an HVAC system in a mid-size commercial building," *HVAC&R Res.*, vol. 20, no. 1, pp. 121–135, 2014.

[34] K. Deng *et al.*, "Model predictive control of central chiller plant with thermal energy storage via dynamic programming and mixed-integer linear programming," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 565–579, Apr. 2015.

[35] I. Sharma *et al.*, "A modeling framework for optimal energy management of a residential building," *Energy Buildings*, vol. 130, pp. 55–63, Oct. 2016.

[36] B. Mayer, M. Killian, and M. Kozek, "A branch and bound approach for building cooling supply control with hybrid model predictive control," *Energy Buildings*, vol. 128, pp. 553–566, Sep. 2016.

[37] S. Salakij, N. Yu, S. Paolucci, and P. Antsaklis, "Model-based predictive control for building energy management. I: Energy modeling and optimal control," *Energy Buildings*, vol. 133, pp. 345–358, Dec. 2016.

[38] R. Zafar, A. Mahmood, S. Razzaq, W. Ali, U. Naeem, and K. Shehzad, "Prosumer based energy management and sharing in smart grid," *Renew. Sustain. Energy Rev.*, vol. 82, pp. 1675–1684, Feb. 2018.

[39] T. Bai, S. Li, and Y. Zheng, "Distributed model predictive control for networked plant-wide systems with neighborhood cooperation," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 1, pp. 108–117, Jan. 2019.

[40] X. Mi and S. Li, "Event-triggered MPC design for distributed systems with network communications," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 1, pp. 240–250, Jan. 2018.

[41] H. Pombeiro, M. J. Machado, and C. Silva, "Dynamic programming and genetic algorithms to control an HVAC system: Maximizing thermal comfort and minimizing cost with PV production and storage," *Sustain. Cities Soc.*, vol. 34, pp. 228–238, Oct. 2017.

[42] F. Ascione, N. Bianco, C. De Stasio, G. M. Mauro, and G. P. Vanoli, "Simulation-based model predictive control by the multi-objective optimization of building energy performance and thermal comfort," *Energy Buildings*, vol. 111, pp. 131–144, Jan. 2016.

[43] S. Wang and X. Jin, "Model-based optimal control of VAV air-conditioning system using genetic algorithm," *Building Environ.*, vol. 35, no. 6, pp. 471–487, 2000.

[44] C. D. Corbin, G. P. Henze, and P. May-Ostendorp, "A model predictive control optimization environment for real-time commercial building application," *J. Build. Perform. Simul.*, vol. 6, no. 3, pp. 159–174, 2013.

[45] D. He, "Dual-mode nonlinear MPC via terminal control laws with free-parameters," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 3, pp. 526–533, Jul. 2017.

[46] K. I. Katsigarakis, G. D. Kontes, G. I. Giannakis, and D. V. Rovas, "Sense-think-act framework for intelligent building energy management," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 31, no. 1, pp. 50–64, 2016.

[47] M. Casals, M. Gangolells, N. Forcada, M. Macarulla, A. Giretti, and M. Vaccarini, "SEAM4US: An intelligent energy management system for underground stations," *Appl. Energy*, vol. 166, pp. 150–164, Mar. 2016.

[48] D. Manjarres, A. Mera, E. Perea, A. Lejarazu, and S. Gil-Lopez, "An energy-efficient predictive control for HVAC systems applied to tertiary buildings based on regression techniques," *Energy Buildings*, vol. 152, pp. 409–417, Oct. 2017.

[49] G. D. Kontes, C. Valmaseda, G. I. Giannakis, K. I. Katsigarakis, and D. V. Rovas, "Intelligent BEMS design using detailed thermal simulation models and surrogate-based stochastic optimization," *J. Process Control*, vol. 24, no. 6, pp. 846–855, 2014.

[50] S. Prívara, J. Cigler, Z. Vána, F. Oldewurtel, C. Sagerschnig, and E. Žáceková, "Building modeling as a crucial part for building predictive control," *Energy Buildings*, vol. 56, pp. 8–22, Jan. 2013.

[51] D. Picard, J. Drgoňa, M. Kvasnica, and L. Helsen, "Impact of the controller model complexity on model predictive control performance for buildings," *Energy Buildings*, vol. 152, pp. 739–751, Oct. 2017.

[52] M. Gruber, A. Trüschel, and J. O. Dalenbäck, "Model-based controllers for indoor climate control in office buildings—Complexity and performance evaluation," *Energy Buildings*, vol. 68, pp. 213–222, Jan. 2014.

[53] S. Petersen and K. W. Bundgaard, "The effect of weather forecast uncertainty on a predictive control concept for building systems operation," *Appl. Energy*, vol. 116, pp. 311–321, Mar. 2014.

[54] F. Oldewurtel, D. Sturzenegger, and M. Morari, "Importance of occupancy information for building climate control," *Appl. Energy*, vol. 101, pp. 521–532, Jan. 2013.

[55] B. Dong and K. P. Lam, "A real-time model predictive control for building heating and cooling systems based on the occupancy behavior pattern detection and local weather forecasting," *Building Simul.*, vol. 7, no. 1, pp. 89–106, 2014.

[56] Y. Kwak, J.-H. Huh, and C. Jang, "Development of a model predictive control framework through real-time building energy management system data," *Appl. Energy*, vol. 155, pp. 1–13, Oct. 2015.

[57] Y. Kwak and J.-H. Huh, "Development of a method of real-time building energy simulation for efficient predictive control," *Energy Convers. Manag.*, vol. 113, pp. 220–229, Apr. 2016.

[58] P. de Wilde, "The gap between predicted and measured energy performance of buildings: A framework for investigation," *Autom. Construct.*, vol. 41, pp. 40–49, May 2014.

[59] A. C. Menezes, A. Cripps, D. Bouchlaghem, and R. Buswell, "Predicted vs. actual energy performance of non-domestic buildings: Using post-occupancy evaluation data to reduce the performance gap," *Appl. Energy*, vol. 97, pp. 355–364, Sep. 2012.

[60] N. Li, Z. Yang, B. Becerik-Gerber, C. Tang, and N. Chen, "Why is the reliability of building simulation limited as a tool for evaluating energy conservation measures?" *Appl. Energy*, vol. 159, pp. 196–205, Dec. 2015.

[61] H. Satyavada and S. Baldi, "An integrated control-oriented modelling for HVAC performance benchmarking," *J. Build. Eng.*, vol. 6, pp. 262–273, Jun. 2016.

[62] E. Žáčeková, Z. Váňa, and J. Cigler, "Towards the real-life implementation of MPC for an office building: Identification issues," *Appl. Energy*, vol. 135, pp. 53–62, Dec. 2014.

[63] Z. Afroz, G. M. Shafiullah, T. Urmee, and G. Higgins, "Modeling techniques used in building HVAC control systems: A review," *Renew. Sustain. Energy Rev.*, vol. 83, pp. 64–84, Mar. 2018.

[64] A. Foucquier, S. Robert, F. Suard, L. Stéphan, and A. Jay, "State of the art in building modelling and energy performances prediction: A review," *Renew. Sustain. Energy Rev.*, vol. 23, pp. 272–288, Jul. 2013.

[65] B. Gunay, W. Shen, and G. Newsham, "Inverse blackbox modeling of the heating and cooling load in office buildings," *Energy Buildings*, vol. 142, pp. 200–210, May 2017.

[66] A. Conserva *et al.*, "Energy in time project: Summary of final results," in *Proc. 12th Conf. Sustain. Develop. Energy, Water Environ. Syst.*, 2017, Paper SDEWES2017-0867.

[67] Z. Li, Y. Han, and P. Xu, "Methods for benchmarking building energy consumption against its past or intended performance: An overview," *Appl. Energy*, vol. 124, pp. 325–334, Jul. 2014.

[68] G. F. Lawler and V. Limic, *Random Walk: A Modern Introduction*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[69] R Core Team. (2018). *R: A Language and Environment for Statistical Computing*. [Online]. Available: http://www.r-project.org/

[70] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Statist. Softw.*, vol. 33, no. 1, pp. 1–22, 2010.

[71] M. S. Al-Homoud, "Computer-aided building energy analysis techniques," *Building Environ.*, vol. 36, no. 4, pp. 421–433, 2001.

[72] W. Liang, R. Quinte, X. Jia, and J.-Q. Sun, "MPC control for improving energy efficiency of a building air handler for multi-zone VAVs," *Building Environ.*, vol. 92, pp. 256–268, Oct. 2015.

[73] B. Koffi, A. Cerutti, M. Duerr, A. Iancu, A. Kona, and G. Janssens-Maenhout, "CoM default emission factors for the member states of the European union—Version 2017," European Commission, Joint Research Center (JRC), Brussels, Belgium, Tech. Rep., 2017.

[74] J. Cigler, P. Tomáško, and J. Široký, "BuildingLAB: A tool to analyze performance of model predictive controllers for buildings," *Energy Buildings*, vol. 57, pp. 34–41, Feb. 2013.

[75] J. H. Lim, J. T. Kim, S. H. Cho, and G. Y. Yun, "Development of the adaptive PMV model for improving prediction performances," *Energy Buildings*, vol. 98, pp. 100–105, Jul. 2015.

[76] B. W. Olesen, "Indoor environmental input parameters for the design and assessment of energy performance of buildings," *REHVA J.*, pp. 17–23, Jan. 2015.

[77] T. Ekwevugbe, N. Brown, V. Pakka, and D. Fan, "Improved occupancy monitoring in non-domestic buildings," *Sustain. Cities Soc.*, vol. 30, pp. 97–107, Apr. 2017.

[78] A. Capozzoli, M. S. Piscitelli, A. Gorrino, I. Ballarini, and V. Corrado, "Data analytics for occupancy pattern learning to reduce the energy consumption of HVAC systems in office buildings," *Sustain. Cities Soc.*, vol. 35, pp. 191–208, Nov. 2017.

[79] Q. Meng, M. Mourshed, and S. Wei, "Going beyond the mean: Distributional degree-day base temperatures for building energy analytics using change point quantile regression," *IEEE Access*, vol. 6, pp. 39532–39540, 2018.

[80] C. Wang, B. Jiao, L. Guo, Z. Tian, J. Niu, and S. Li, "Robust scheduling of building energy system under uncertainty," *Appl. Energy*, vol. 167, pp. 366–376, Apr. 2016.

[81] H. Thieblemont, F. Haghighat, R. Ooka, and A. Moreau, "Predictive control strategies based on weather forecast in buildings with energy storage system: A review of the state-of-the art," *Energy Buildings*, vol. 153, pp. 485–500, Oct. 2017.

[82] E. Feller, L. Ramakrishnan, and C. Morin, "Performance and energy efficiency of big data applications in cloud environments: A Hadoop case study," *J. Parallel Distrib. Comput.*, vols. 79–80, pp. 80–89, May 2015.

[83] J. Gómez-Romero, M. Molina-Solana, M. Ros, M. D. Ruiz, and M. J. Martin-Bautista, "Comfort as a Service: A new paradigm for residential environmental quality control," *Sustainability*, vol. 10, no. 9, p. 3053, Aug. 2018.

[84] F. Ferracuti *et al.*, "Data-driven models for short-term thermal behaviour prediction in real buildings," *Appl. Energy*, vol. 204, pp. 1375–1387, Oct. 2017.

**JUAN GÓMEZ-ROMERO** received the degree in computer science and the Ph.D. degree in intelligent systems from the Universidad de Granada, in 2004 and 2008, respectively.

He was a Lecturer with the Applied Artificial Intelligence Group, Universidad Carlos III de Madrid, from 2008 to 2013, and a Research Associate in the EU FP7 Project Energy IN TIME with the Universidad de Granada, from 2013 to 2017. He was also a Visiting Researcher with the Data Science Institute, Imperial College London, from 2016 to 2017. He has been a Senior Research Fellow with the Computer Science and Artificial Intelligence Department, Universidad de Granada, since 2016. He has participated in more than 20 projects in security, ambient intelligence, and energy efficiency. His research interests include the use of semantic representation models and machine learning techniques to perform automatic reasoning towards higher-level information fusion.

Dr. Gómez-Romero is the Principal Investigator of the projects BIGFUSE: Semantics for Big Data Fusion and Analysis: Improving Energy Efficiency in Smart Grids and PROFICIENT: Deep Learning for Energy-Efficient Building Control.

**CARLOS J. FERNÁNDEZ-BASSO** received the degree in computer science and the M.Sc. degree in data science from the Universidad de Granada, in 2014 and 2015, respectively, where he is currently pursuing the Ph.D. degree in computer science and energy efficiency.

He was a Lead Developer in the EU FP7 Project Energy IN TIME in the topics of building simulation and control, data analytics, and machine learning. He also collaborates with the Data Science Institute, Imperial College London, where he has carried out research stays, from 2016 to 2018. He is currently a Research Assistant with the Computer Science and Artificial Intelligence Department, Universidad de Granada.

**M. VICTORIA CAMBRONERO** received the degree in industrial engineering with a specialization in construction and industrial facilities and the M.Sc. degree in energy technology for sustainable development from the Polytechnic University of Valencia, in 2008 and 2009, respectively.

She was a Project Developer and a Researcher on building management and energy efficiency with the Institute of Energy Engineering, Valencia, before joining Acciona Infrastructures R&D Department, in 2010. Since 2018, she has been a Project Manager with Acciona Ingeniería. She has a wide experience in research and innovation projects and has collaborated and managed several FP7 and H2020 projects in renewable energies, HVAC systems and storage integration, and control strategies for energy efficiency improvement in buildings and districts; e.g., 2DISTRICT, EnergyINTIME, LowUP, Flexynets, FC-DISTRICT, MESSIB, EINSTEIN, CommONEnergy, and COST-EFFECTIVE. She is currently an Industrial Engineer specialized in energy efficiency and building simulation.

Ms. Cambronero holds a Project Management Professional certification granted by the Project Management Institute and a Certified in Measurement and Verification Protocol certification granted by the Association of Energy Engineers and Efficiency Valuation Organization.

**MIGUEL MOLINA-SOLANA** received the degree in computer science and the Ph.D. degree from the Universidad de Granada, in 2007 and 2012, respectively.

From 2012 to 2015, he was a Research Associate with the Universidad de Granada in the FP7 Project Energy IN TIME. He was a Research Associate on visualization with the Data Science Institute, Imperial College. He is currently a Marie Curie Research Fellow with Imperial College London, U.K. His research interests include applied work in machine learning and knowledge representation in diverse domains such as music, energy management, and business.

Dr. Molina-Solana is the Principal Investigator of the H2020 Project DATASOUND–Understanding Data With Sound.

**JESÚS R. CAMPAÑA** received the M.Sc. and Ph.D. degrees in computer science from the Universidad de Granada, where he has been a member of the Intelligent Databases and Information Systems Research Group, Department of Computer Science and Artificial Intelligence, since 2005.

From 2013 to 2018, he was a Lecturer with the Universidad de Granada. He has been a member of several research projects related to fuzzy data representation in databases, data mining, text mining, and energy efficiency, including the FP7 Project Energy IN TIME. His research interests include fuzzy databases, XML, knowledge representation, semantic web, data mining, and text mining.

**M. DOLORES RUIZ** received the degree in mathematics and the European Ph.D. degree in computer science from the Universidad de Granada, in 2005 and 2010, respectively.

She held a non-permanent teaching positions with the Universities of Jaén, Granada, and Cádiz. She has participated in more than ten projects, including the EU FP7 Projects ePOOLICE and Energy IN TIME. She is currently a Research Associate with the Computer Science and Artificial Intelligence Department, Universidad de Granada. Her research interests include data mining, information retrieval, energy efficiency, big data, correlation statistical measures, sentence quantification, and fuzzy sets theory. She has organized several special sessions about Data Mining in international conferences and was part of the organization committee of the FQAS'2013 and SUM'2017 conferences.

Dr. Ruiz belongs to the Approximate Reasoning and Artificial Intelligence Research Group and the Cybersecurity Lab, Universidad de Granada. She has been the Principal Investigator of the project Exception and anomaly detection by means of fuzzy rules using the RL-theory. Application to fraud detection.

**MARIA J. MARTIN-BAUTISTA** received the degree in computer science and the Ph.D. degree from the Universidad de Granada, in 1996 and 2000, respectively.

She has been a Professor with the Computer Science and Artificial Intelligent Department, Universidad de Granada, since 2018. She was the Principal Investigator in the FP7 European Project Energy IN TIME with the Universidad de Granada, from 2013 to 2017. She has also been a Principal Investigator of several international and national projects and knowledge transfer contracts with private companies. She has published more than 100 papers in international journals and conferences. Her current research interests include intelligent systems, big data, and knowledge representation with applications to energy, security, and health.

Prof. Martin-Bautista is a member of the IEEE Society and the EUSFLAT Society.

## 5.3 A fuzzy mining approach for energy efficiency in a Big Data framework (Published in IEEE Transactions on Fuzzy Systems)

- Carlos Fernandez Basso, M. Dolores Ruiz, Maria J. Martin-Bautista. IEEE transactions on Fuzzy Systems (2020).
  - Status: **Published**.
  - Impact Factor (JCR 2018): **8.759**
  - Subject Category: **Computer Science, Artificial Intelligence**
  - Rank: **6/134**
  - Quartile: **Q1**

# A fuzzy mining approach for energy efficiency in a Big Data framework

Carlos Fernandez-Basso, M. Dolores Ruiz and Maria J. Martin-Bautista, *Member, IEEE*

***Abstract*—The discovery and exploitation of hidden information in collected data has gained attention in many areas, particularly in the energy field due to their economic and environmental impact. Data mining techniques have then emerged as a suitable toolbox for analysing the data collected in modern network management systems in order to obtain a meaningful insight into consumption patterns and equipment operation. However, the enormous amount of data generated by sensors, occupational and meteorological data involve the use of new management systems and data processing. Big Data presents great opportunities for implementing new solutions to manage these massive data sets. In addition, these data present values whose nature complicates and hides the understanding and interpretation of the data and results. Therefore the use of fuzzy methods to adequately transform the data can improve their interpretability. This paper presents an automatic fuzzification method implemented using the Big Data paradigm, which enables, in a later step, the detection of interrelations and patterns among different sensors and weather data recovered from an office building.**

***Index Terms*—Data mining, big data, fuzzy association rules, energy building, operational research.**

## I. Introduction

NOWADAYS, modern management systems can generate thousands of measurements every minute from the different detection devices that record their operation. Companies in the energy sector are increasingly aware of the great opportunity that the analysis and exploitation of these data can bring (see for instance [1], [2]). To this end, there is a tendency to store and process this type of data in order to obtain a meaningful insight into consumption patterns and the operation of equipment.

However, the enormous amount of data generated by sensors and other data sources, such as meteorological and occupancy data, require new infrastructures and algorithms capable of storing, processing and analysing them. Big Data presents great opportunities for implementing new solutions to manage these massive data sets. Moreover, the nature of these data can be diverse and can be described in numerical, categorical, imprecise forms. To improve the interpretability of the data we can modify the knowledge extraction algorithm through the use of fuzzy logic to create, for example, linguistic labels for sensors with numerical values that also provide meaningful semantics for the user.

The main challenge when processing large energy data is therefore to provide adequate methods and techniques capable of improving the quality of the data generated by the sensor metering of buildings. In this regard, different methods can be applied during the pre-processing phase in order to detect outliers [3] or by applying other cleaning data procedures. In general, these data are massive because they are generated with low frequencies by a large number of sensors. This massive quantity gives grounds to use Big Data techniques to process the data in a distributed way, thus improving the efficiency of the processes.

Sensor metering data are very often collected by means of numerical measurements taking values within a continuous range. This increases the difficulty of analysing them on a larger scale due to their fine granularity. A primary approach is to divide the range of possible values into intervals in order to help the algorithms to process the data. But this division suffers from some drawbacks: firstly, the results can vary a lot depending on the applied division, and secondly, this division may not be very intuitive for its later analysis of results. Fuzzy sets have been proven to adequately represent data with soft borders, increasing the interpretability of results by associating meaningful linguistic labels to the generated fuzzy sets. Other approaches use interval programming methods to tackle the uncertainty that the data may contain [4].

In this paper we propose a fuzzification algorithm to adequately pre-process the data in order to apply, in a later step, fuzzy data mining techniques to discover potentially useful information that maybe hidden in the data.

In particular, we have applied association-rule discovery, an unsupervised technique able to find existing relationships between variables and their values. In addition, these results allow the operators to carry out procedures and the use of the equipment in the building better, thus improving its energy efficiency. On the other hand, the use of weather and occupation patterns for the energy use of the building would allow the optimization of the needs of the building in specific situations.

To this end, we have deployed the whole system following the Map-Reduce paradigm which allows the distributed computation of large volumes of data. Specifically, we used the Spark platform [5] together with an unstructured storage following NoSQL specifications, which enables an efficient storage of sensor data collected in buildings.

The whole system has been successfully applied in an office building located in Bucharest, obtaining a set of patterns describing the operational and energetic functioning of the building. Nevertheless, the presented approach could also be applied in other types of buildings.

The obtained patterns describe the day-to-day working of the building, but they could also help to discover the poor functioning of some systems due to abnormal circumstances.

C. Fernandez-Basso and M.J. Martin-Bautista are with the Department of Computer Science and A.I. and CITIC-UGR, and M.D. Ruiz is with the Department of Statistics and Operative Research, both from the University of Granada, Spain (e-mail: {cjferba, mdruiz, mbautis}@decsai.ugr.es).

The work is structured as follows. The next section reviews previous related research and introduces the necessary background of related concepts. Section III describes the design of our system focusing on the developed fuzzification algorithm. Section IV presents our results for the office building located in Bucharest. Finally, in Section VI we summarize the conclusions and present future prospective research lines.

## II. Background

Data mining techniques are widely used in the field of energy as can be observed in [6]–[9]. In [6], [7] the authors reviewed how some of the traditional data mining techniques have been used to obtain construction-related information. Additionally, an association-rule discovery tool was used to explore correlations between building data in two different time periods: a day and a year. This study allowed the detection of some equipment failures in two ventilation units and to propose low-cost strategies for saving energy. In [8], he authors focused on the different data mining techniques used for energy management, especially in the construction sector, discussing the main challenges and opportunities that will arise with the advent of new computational technologies such as Big Data.

A more recent revision can be found in [9] where unsupervised data mining techniques are presented, paying special attention to the operational data mining of massive data collected from buildings in order to find significant patterns. More recent reviews such as [10], focus on the use of data mining tools to predict the consumption of the building by taking into account sensor metering, time and building occupancy data.

There are very few studies which have investigated fuzzy data in the field of energy management. In this regard, we can highlight the research in [11] where the authors look for anomalies in sensor data under uncertainty.

Our research, is in a similar direction but we propose to automatically fuzzify data collected by sensors and afterwards apply fuzzy association rule mining to discover potentially useful patterns in the field of energy management in buildings.

### A. Our approach

In Figure 1 depicts the complete system of our proposal. It can be divided into three big blocks. In the first step, the data collection is carried out. Depending on the building different types of data are generated with more or less reliability. For example, in the case of non-residential buildings (the example of our case study) the agenda containing the occupation of the building will be more reliable than that of a hotel, where the guests do not have fixed schedules.

The second step comprises the core of the computation with different phases: pre-processing, fuzzification and application of data mining techniques. In our approach, distributed processing tools are used to improve efficiency and computational capacity.

Once we have obtained the results of our analytical techniques, the last step is the interpretation of results by experts or operators of the system in order to obtain knowledge that maybe useful for improving the maintenance processes and the energy efficiency of the building.

This whole process has been applied in the field of the Energy IN TIME project to non-residential buildings: a hotel, an airport and two office buildings, although we just present here the results obtained from one of the office buildings. Nevertheless, the proposed system is general and can be applied to other types of buildings.

### B. Fuzzy Association Rules

In the data mining field, association rules are used to discover facts that often occur together within a particular dataset. A typical example of this type of problem is figuring out which products from a supermarket are normally bought together. Association rules were formally defined for the first time by Agrawal et al. [12], although the analysis of associations was investigated much earlier in the more general framework of Observational Calculi in [13]. The problem consists of discovering implications of the form $A \rightarrow B$ where $A, B$ are subsets of items from $I = \{i_1, i_2, \ldots, i_m\}$ fulfilling that $A \cap B = \emptyset$ in a database formed by a set of $n$ transactions $D = \{t_1, t_2, \ldots, t_n\}$ each of them containing subsets of items from $I$. $A$ is usually referred as the antecedent and $B$ to the consequent of the rule.

The problem of discovering association rules is divided into two sub-tasks:

- Finding all the itemsets above the minimum support threshold, where support is provided by the percentage of transactions containing the items. These sets of items, or itemsets, are known as frequent itemsets.
- On the basis of the found frequent itemsets, rules are discovered as those exceeding the minimum threshold for confidence or another assessment measure generally established by the user.

However, the nature of the data can be diverse and can be described numerically, categorically or imprecisely. In the case of numerical elements, a first approximation could be to categorise them so that, for example, the temperature of a room can be given by a range to which it belongs, such as [24°, 30°]. However, depending on how these intervals are defined, the results obtained can vary a lot. To avoid this, the use of linguistic tags such as “warm” represented by a fuzzy set is a good option to represent the temperature of a room, having at the same time significant semantics for the user [14]. Beside this, we may also have a dataset with inherent imprecise knowledge where ordinary crisp methods cannot be directly applied (see for instance [15]).

To deal with this kind of data the concept of fuzzy transaction and fuzzy association rule are defined in [16], [17].

*Definition 1:* Let $I$ be a set of items. A fuzzy transaction, $t$, is a non-empty fuzzy subset of $I$ in which the membership degree of an item $i \in I$ in $t$ is represented by a number in the range [0, 1] and denoted by $t(i)$.

By this definition a crisp transaction is a special case of fuzzy transaction. We denote by $\tilde{D}$ a database consisting in a set of fuzzy transactions.

*Definition 2:* Let $A \subseteq I$ be an itemset, i.e. a subset of items in $I$. The degree of membership of $A$ in a fuzzy transaction

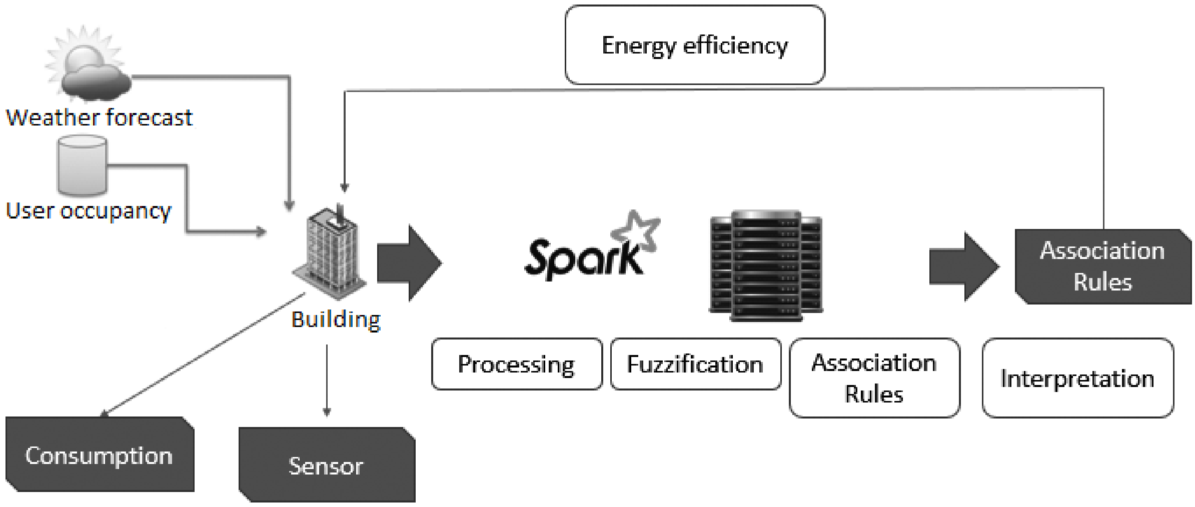


Fig. 1. General process of our proposal

$t \in \tilde{D}$ is defined as the minimum of the membership degree of all its items:

$$t(A) = \min_{i \in A} t(i) \tag{1}$$

*Definition 3:* Let $A, B \subseteq I$ be itemsets in the fuzzy database $\tilde{D}$. Then, a fuzzy association rule $A \rightarrow B$ is satisfied in $\tilde{D}$ if and only if $t(A) \leq t(B)\, \forall t \in \tilde{D}$, that is, the degree of satisfiability of $B$ in $\tilde{D}$ is greater than or equal to the degree of satisfiability of $A$ for all fuzzy transactions $t$ in in $\tilde{D}$.

Assessment measures for fuzzy association rules have been studied in numerous papers according to different perspectives (see a review in [18]). The approach followed here it is a cardinality based generalization presented in [17] and extended in other works (see for instance [19], [20]) by considering a finite set of $\alpha$-cuts for the unit interval.

*Definition 4:* The support of an itemset $A$ in a fuzzy database $\tilde{D}$ is defined as:

$$FSupp(A) = \sum_{\alpha_i \in \Lambda} (\alpha_i - \alpha_{i+1}) \frac{|\{t \in \tilde{D} : t(A) \geq \alpha_i\}|}{|\tilde{D}|} \tag{2}$$

where $\Lambda = \{1 = \alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of $\alpha$-cuts.

*Definition 5:* The support of a fuzzy rule $A \rightarrow B$ in a fuzzy database $\tilde{D}$ is defined as:

$$FSupp(A \rightarrow B) = \sum_{\alpha_i \in \Lambda} (\alpha_i - \alpha_{i+1}) \frac{|\{t \in \tilde{D} : t(A) \geq \alpha_i \text{ and } t(B) \geq \alpha_i\}|}{|\tilde{D}|} \tag{3}$$

where $\Lambda = \{1 = \alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of $\alpha$-cuts.

*Definition 6:* The confidence of a fuzzy rule $A \rightarrow B$ is defined as:

$$FConf(A \rightarrow B) = \sum_{\alpha_i \in \Lambda} (\alpha_i - \alpha_{i+1}) \frac{|\{t \in \tilde{D} : t(A) \geq \alpha_i \text{ and } t(B) \geq \alpha_i\}|}{|\{t \in \tilde{D} : t(A) \geq \alpha_i\}|} \tag{4}$$

where $\Lambda = \{1 = \alpha_1, \alpha_2, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ and $\alpha_{p+1} = 0$ is a set of $\alpha$-cuts.

By using support and confidence measures and setting appropriated thresholds, fuzzy association rules can be discovered by fixing a set of predefined $\alpha$-cuts [19]. Note that considering a sufficiently dense set of $\alpha$-cuts in the unit interval, the obtained measure will be a good approximation of the real measure that should consider every $\alpha \in [0, 1]$ appearing in the dataset.

### C. Big Data paradigm

The most famous framework for Big Data is *MapReduce* designed by Google in 2003 [21]. It has become one of the most relevant tools for processing large datasets with parallel and distributed algorithms in a cluster. The MapReduce framework manages all data transfers and communications amongst the systems. It also provides redundancy, fault-tolerance and job scheduling. In this programming paradigm we usually have two phases. Firstly the $Map()$ function, which makes the processing of data and returns the data transformed into key value pairs depending on our necessities. Secondly, the $Reduce()$ function which aggregates the lists of $< key, value >$ pairs sharing the same key to obtain a piece of processed data.

MapReduce algorithms can be programmed in different frameworks. One of the most used that have been proven and which works quite fast is Apache Spark [22]. It appeared as an open-source framework built around speed, ease of use, and sophisticated analytics [5]. The most important feature of Spark is that it allows in-memory computing, and, as a consequence more complex algorithms can be developed. This is because Spark supports an advanced Directed Acyclical Graphics (DAG) execution engine that allows more complex data flows using several MapReduce phases, a procedure that is not possible with other tools such as Hadoop.

Apache Spark has implemented a data structure to abstract the concept of data partition. This structure is called the Resilient Distributed Dataset (RDD) [23], meaning that data are distributed across the clusters. The RDD has two different types of operations. The first type of transformation converts the RDD structure into a different RDD, which is called *Transformation operations*. The second type is *evaluation Actions* performed over the above transformations which return a final value for each RDD partition. The programmer has to take into account that evaluations are not executed until a specific *Action operation* is specified in the code. This is due to the "lazy" evaluation of Spark that strongly distinguishes between Transformations and Actions.

For the implementation of the proposed methodology the Spark tool has been used due to its large computing capacity and compared to Hadoop because it uses memory storage, thus improving its efficiency (see the complete comparison made in [22] where Hadoop and Spark frameworks are compared in several Machine Learning algorithms).

## III. Methodology

In our proposal, before applying fuzzy association-rule mining, we have to pre-process the data collected from the building. We can observe the workflow of the proposal in Figure 2, where we have distinguished several phases.

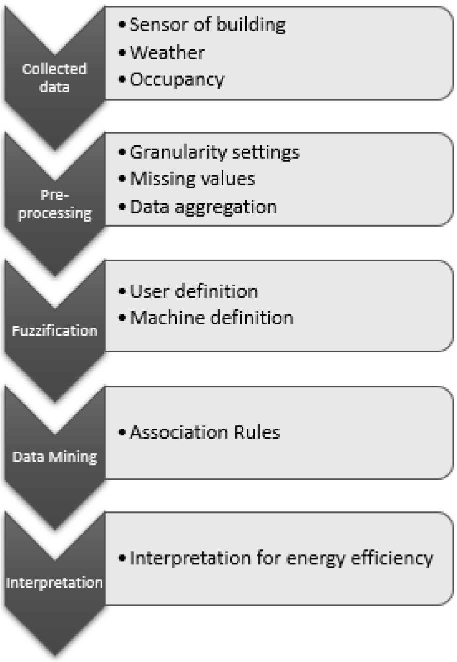


Fig. 2. General workflow of the proposed data mining framework

The first phase comprises the collection and storage of building data which are e.g. data from sensors and building equipment, weather and occupation. All these data are stored in a NoSQL database MongoDB [24] for their efficient management in the insertion and search of documents. Moreover, this MongoDB enables the storage of data with different structures and different fields [25]. This choice is due to the fact that the data from the sensors arrive with a very low frequency so the insertions in the database must be fast, and also each sensor can have values with different structures (e.g. XML, real values, strings, etc.)

Subsequently, the pre-processing phase of the data is carried out. The collected data from sensors may contain lost values, outliers and so on and therefore some transformation to be used by the algorithms is needed.

After this, the continuous numerical values are transformed following a fuzzification method explained in III-C. By means of this process the range of values are divided into meaningful fuzzy sets where some linguistic labels can be associated to each fuzzy set. This step improves the interpretability of the data and the obtained results will adjust better to the nature of the variables. Depending on the type of variable, the fuzzification procedure can either be carried out automatically or by the help of expert knowledge.

Finally, when we have the pre-processed and transformed the data, we proceed to apply data mining techniques. In this case study, extraction of fuzzy association rules has been applied in order to extract hidden relationships from the sensors, occupation and the environment of the building. Once the results have been obtained, the discovered patterns are interpreted with the help of end users to improve the energy efficiency of the building.

### A. Data collection

The developed methodology has been used in a large office building in Bucharest. In Figure 3 the different data sources collected and added to the database can be seen.

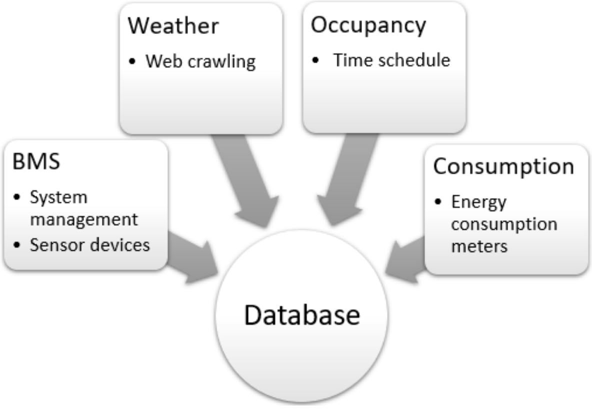


Fig. 3. Schema of the type of data collected for the office building

Each of these pieces of data were collected using different procedures. On the one hand, data provided by Building Management Systems (BMS) [26] such as sensors, actuators and building equipment data have been considered. In particular, sensors and BMS data have been sent via a real-time messaging system API (rabbitmq [27]) to the database. This includes energy consumption metering from the building. On the other hand, the occupancy data have been obtained by processing the working hours of the offices in the building. Finally the weather has been obtained using a web crawler that stores, in a structured way, the meteorological forecast foreach day.

### B. Pre-processing

Different techniques are commonly used in data preprocessing. First of all, the granularity of the data is standardized. Since the data from the sensors are collected in different instants of time it is necessary to group these measurements in the same instants of time. Table 1 shows an example of this transformation. In the example the values of the time instants are grouped every 15 minutes (this parameter can be changed according to user preferences). Depending on the type of values the method used can be fixed according to different criteria. For instance, for continuous data we may be interested in considering the average (as in the example of Table I) or consider the last value sent by the sensor. This transformation allows us to generate a set of transactional data suitable for applying data mining techniques.

TABLE I
EXAMPLE OF THE TEMPORAL GRANULARITY PROCESSING OF SOME TEMPERATURE DATA

| Time | Temperature |
|---|---|
| 1/1/2016 15:14:30 | 16 |
| 1/1/2016 15:16:45 | 17 |

↓

| Time | Temperature |
|---|---|
| 1/1/2016 15:15:00 | 16.5 |

Some of the collected data come from energy meters, which function as accumulators. You can see an example of this type of data in Figures 4 and 5 where its value is always increasing and represents the energy consumption can be seen. This type of variable has been processed through a transformation function to represent the consumption of the building in different time slots . That is, each temporal piece of data represents the consumption in that interval of time. We can see an example after applying this transformation in Figure 5.

After processing the data and obtaining the desired granularity, transactions with lost values are eliminated, as well as transactions containing outlier measurements made by sensors. The *sklearn* python library [28] was used to determine these outliers, specifically using the elliptic envelope fitting function [29]. At the end of this procedure, the transactional database comprises sensor, weather and occupation data for each fraction of time without anomalous values.

### C. Fuzzification

The collection of data from sensors, counters, weather or building occupation variables has a nature that is difficult to represent and interpret by end users since they are continuous values with often complex measures to interpret and understand. Fuzzification of these data can improve the results found by the mining algorithms, and at the same time increase the interpretability of the results.

We propose a fuzzification algorithm that allows an automatic processing defined by the machine by using the data values according to their distribution. In addition, the algorithm allows the definition of the fuzzy labels by a user (see the two different types of input that can be provided in Algorithm 1). For this, we have developed a distributed algorithm in Spark following the MapReduce philosophy. This enables the processing of large amounts of data, as in the case of sensor generated data in buildings. The general process is described in Algorithm 1. For this we used Spark for the distribution of data throughout the cluster. The algorithm has as input a dataset, a python dictionary (hash) and an integer. The dictionary is used to store the intervals and labels for variables that have been defined by experts. On the other hand, the default number of tags is used for variables that have not been defined by users and will be automatically created depending on their distribution. Note that Spark automatically divides data into chunks for distributed calculation. We have specified this with the acronym DCS (distribute computing using Spark) and representing each piece of data by $S_i$. In line 6 a global variable is used throughout the whole cluster can be seen, which is then used by the function that distributes the computation through MapReduce (line 8 of Algorithm 1).

**Algorithm 1** Main Spark procedure for Fuzzification preprocessing algorithm

```
1: Input: Data: RDD transactions: ${t_1, ..., t_n}$
2: Input: DefaultIntervals: number of intervals automatically generated by the algorithm
3: Input: Intervals: Hash-list of intervals for each variable: $\{Variable_i : [\{Intervals\}, \{Labels\}], \ldots, Variable_p : [\{Intervals\}, \{Labels\}]\}$
4: Output: Fuzzy transactions containing fuzzified values
                         Start Algorithm
5: Features = Dataset.NameFeatures()
6: $broadcast(Global\_Features)$ #Create a broadcast variable for its use across the cluster
7: DCS in $q$ chunks of Data: $\{S_1, \ldots, S_q\}$
8:     $FuzzyData_{S_i} \leftarrow S_i$.Map $(Fuzzification(t_k \in S_i))$
       # Map function computes independently each transaction in $S_i$
9:     $FuzzyDatabase$ =
       = ReduceByKey$(Aggregation(FuzzyData_{S_1}, \ldots, FuzzyData_{S_q}))$
10: return $FuzzyDatabase$
```

Additionally, in line 8 the procedure calls the fuzzification function described in Algorithm 2. This function is divided into different parts. Firstly, it checks if the name of the variable is found in the $Intervals$ hash-list, if it is found in the python dictionary the new fuzzified variables are created using the names of the labels specified by $Intervals$ and its configuration (i.e. computation of membership degrees) attending to the specified interval (see lines 10-16 of Algorithm 2). If the variable is not found in the dictionary, an automatic procedure is used that divides the values of the variable in a number of intervals defined in $DefaultIntervals$ according to the percentiles of the variable. Figure 6 presents an example with the value of $DefaultIntervals = 3$ where the $y$-axis represents the degree of membership and $x$-axis the percentile of the variable. In this example, percentiles employed have been 25 and 37.5 for defining the trapezoidal form of the first label and left part of second label, and, 62.5 and 75 to define the right part of second label and the third label. So, the $GenerateIntervals$ function divides the set into $k$ equidistributed fuzzy sets using the corresponding percentiles. For instance, for $k = 4$, the considered percentiles are computed as follows:

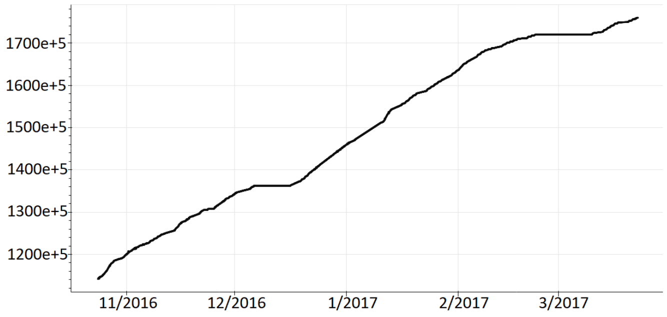


Fig. 4. Heating consumption counter

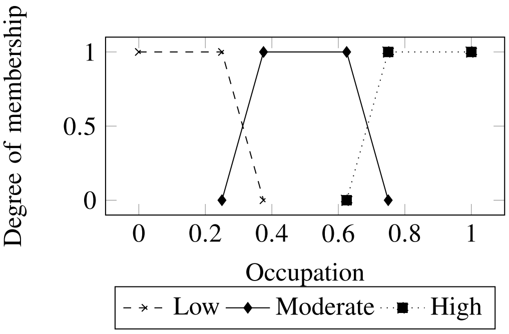


Fig. 6. Example of automatic execution with 3 default intervals

$$\{\frac{100}{k+1}, \frac{100}{k+1} + \frac{100}{(k+1)(k-1)}, \frac{2\cdot 100}{k+1} + \frac{100}{(k+1)(k-1)},$$

$$\frac{2\cdot 100}{k+1} + \frac{2\cdot 100}{(k+1)(k-1)}, \frac{3\cdot 100}{k+1} + \frac{2\cdot 100}{(k+1)(k-1)},$$

$$\frac{3\cdot 100}{k+1} + \frac{3\cdot 100}{(k+1)(k-1)}\}$$

which results in

$$\{p_{20}, p_{26.6}, p_{46.6}, p_{53.3}, p_{73.3}, p_{80}\}$$

On the contrary, the $FuzzyDivision$ function uses the defined intervals contained in the global variable $Intervals$.

For the use case under study, the experts determined different intervals for generating the fuzzy labels. These depend on the nature of the variable, e.g. external temperature, humidity, occupation, etc. Examples of these fuzzy intervals can be seen in Figures 10, 12, 11, 13. Figures 7, 8, 9 show an example of the heating consumption data (see Figure 4 before fuzzification) after their transformation into three different fuzzy sets with the labels: Low, Medium and High.

### D. Data mining:Fuzzy Association Rules

After data pre-processing and fuzzification, data mining techniques were applied to the processed data. In particular, an algorithm for association-rule discovery was applied in Big Data (BDFARE Apriori-TID Big Data Fuzzy Association-Rule Extraction [30], [31]). This algorithm was also implemented following the MapReduce paradigm under the Spark Framework and enables the processing of huge sets of fuzzy transactions, finding frequent itemsets and fuzzy association rules exceeding the imposed thresholds for support and confidence, given a set of $\alpha$-cuts.

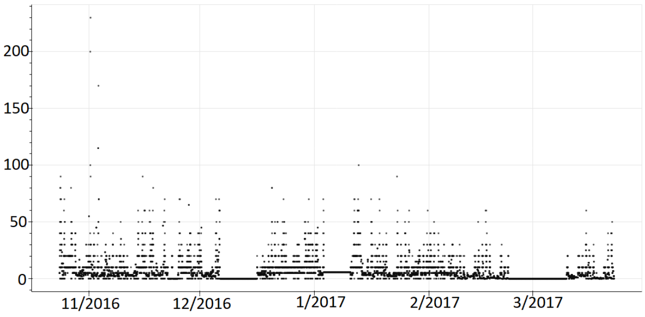


Fig. 5. Heating consumption (every 15 min)

**Algorithm 2** Fuzzification function

```
Input: Data: A transaction: $t_k = \{item_1, \ldots, item_m\}$
Global distributed variable: Intervals: Hash-list of intervals for each variable : $\{Variable_1 : [\{Intervals\}, \{Labels\}], \ldots, Variable_p : [\{Intervals\}, \{Labels\}]\}$
Input: DefaultIntervals: number of intervals automatically generated by the algorithm
Output: Fuzzy transaction
                         Start Algorithm
Features = Dataset.NameFeatures()
DistributeVariable(Features)
DCS in $q$ chunks of Data: $\{S_1, \ldots, S_q\}$
i=0
do
    # Check if the variable exists in the hash list
    if Feature[i] $\in$ Intervals then
        Interval=Intervals[Feature[i][0]]
        Labels=Intervals[Feature[i][1]]
    else
        Interval = GenerateIntervals(DefaultIntervals,Data[Feature[i]])
        Labels = GenerateLabels(DefaultIntervals)
    end if
    for $j = 0; j < |Labels|; j$++ do
        FuzzyData[Label]=FuzzyDivision(Interval[j], Interval[j+1], type)
         # type ="linear", "exponential", "logarithmic"...
        i++
    end for
while $|Feature| > i$
return $FuzzyData$
```

## IV. Results

The results from applying our proposal must be analysed from two points of view. On the one hand, the efficiency and capabilities of our distributed processing that allows the more efficient processing of larger datasets. On the other hand, by means of this processing, fuzzification of the variables and their application to discover fuzzy association rules. The aim is to extract energy patterns in order to improve the knowledge we have about the functioning of the building and to be able to improve tasks such as maintenance and energy efficiency by means of an improvement in the use of the systems.

Data collected from an office building in Romania were retrieved for the analysis. The building is located in Bucharest,

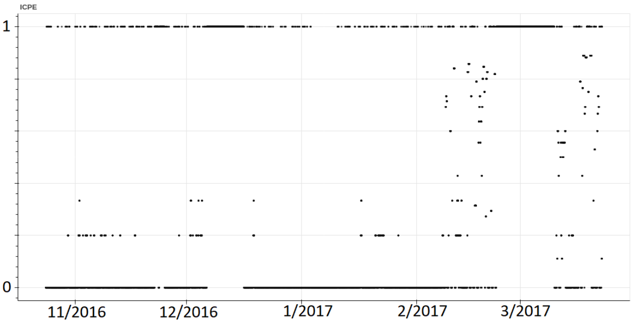


Fig. 7. Low fuzzy label (heating consumption)

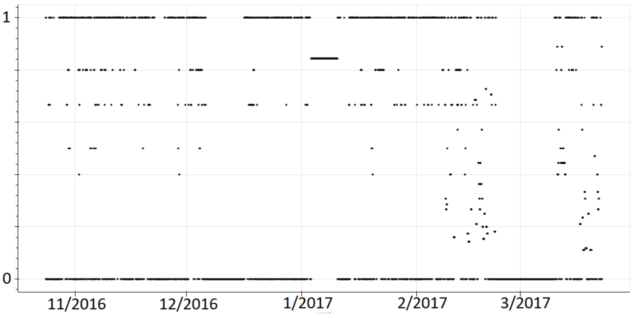


Fig. 8. Medium fuzzy label (heating consumption)

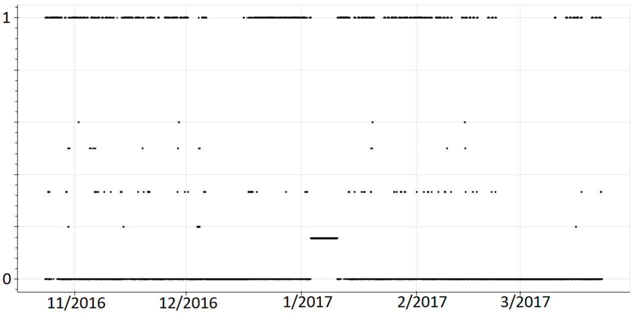


Fig. 9. High fuzzy label (heating consumption)

a city with warm summers and very cold and dry winters. The building is comprised of offices with a constant flow of people and fixed-scheduled plans for indoor conditions.

The considered set of data comprises 273 sensors containing different metering data with a total of 3,649,678 transactions corresponding to data collected from September 2016 to September 2017. The procedure to collect the data by the system was described in section III-A. The set of sensors can be roughly classified into meters and sensor status. More specifically we distinguished the following groups: (1) electric energy, (2) heating agent, (3) domestic water, (4) air-conditioning, (5) temperature, and (6) humidity meters. From the setup and status category there are different sensors related to heating, lightning, windows, etc.

### A. *Efficiency Analysis*

As previously mentioned, the proposal was implemented using Spark which enables MapReduce implementation in large data sets. We have carried out different tests in order to be able to analyse the improvement obtained by processing and fuzzifying these data using this framework. The experimental evaluation have been made on a 64-bits architecture server with 16 cores on 2 Intel Xeon E5 and with 120 GB of RAM, functioning over an operative system with CentOS 6.8. Disabling intel's hyperthreading functionality for testing. The Spark version was 2.2 using a fully distributed mode with Cloudera Manager. DDepending on the number of processors, different percentages of improvement can be achieved (regarding the computation time). In Figure 14 the improvement achieved with different configurations can be seen (2, 6 and 12 cores).

With the purpose of analysing the *speed up* and the *efficiency* [32]–[34] according to the number of cores, we have employed the known measure of speed up defined as [34], [35]

$$S_n = T_1/T_n \tag{5}$$

where $T_1$ is the time of the sequential algorithm and $T_n$ is the execution time of the parallel algorithm using several cores. The efficiency [32]–[34] can be defined in a similar way as

$$E_n = S_n/n = T_1/(n \cdot T_n) \tag{6}$$

Figures 15 and 16 show that the efficiency and speedup are improved as the number of cores increases, even if they are not optimal. The decrease in the efficiency is due to the core workloads and the network congestion used for communication amongst the cores.

In addition, Figure 15 shows the speedup and evolution of the execution times consumed by the proposal. In this figure, it is clearly observed that the greatest reduction in calculation time is achieved when the number of processors is 12.

### B. *Knowledge discovered*

The experiments have been applied for different threshold configurations. In particular, we show here the results obtained when the minimum support was set to 0.6 and minimum confidence to 0.8. We also considered a set of ten equidistributed $\alpha$-cuts. The support and confidence thresholds have been set higher than usual due to the high number of resulting rules obtained for lower values. In Figure 17, the relationship amount of the support and the number of rules obtained is shown. As can be observed, the number of rules increases as the support increases. Figure 18 shows the distribution of the obtained rules (for an experiment with a support of 0.6) according to their confidence.

Figures 18 and 20 show a summary of the quantity of association rules found, taking into account their support and confidence with the mentioned configuration Figure 22 shows a sub-set of the discovered rules in the form of matrix, with the consequent (LHS) and the antecedent (RHS) of the rules.

### C. *Interpretation of the results*

The obtained set of rules has allowed us to discover hidden patterns in the operation of the building, which experts can

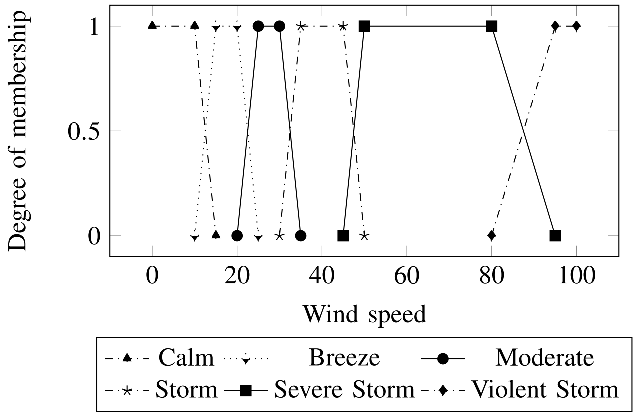


Fig. 10. Wind Speed

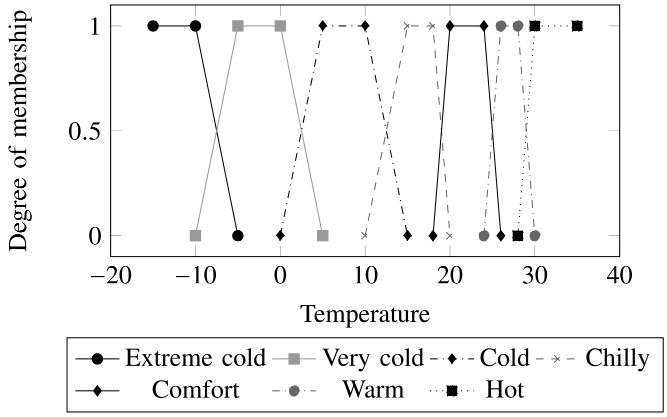


Fig. 11. Temperature

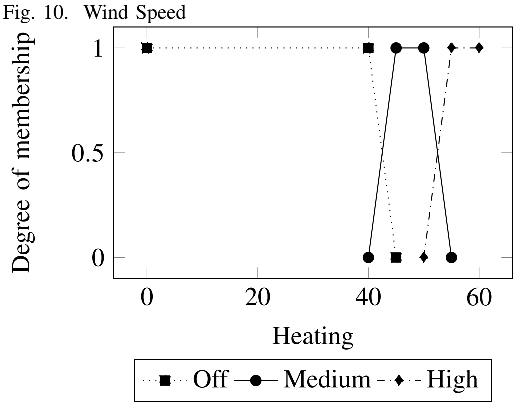


Fig. 12. Heating

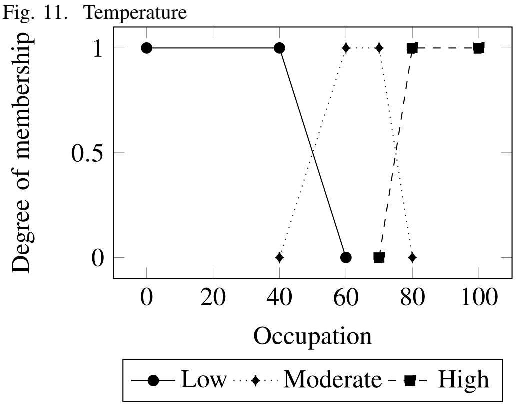


Fig. 13. Occupation

then use to improve its efficiency and maintenance.

Having a look at the discovered patterns we can highlight different rules. For example, in Figure 22 the rule at the top left of the graph (position column 3, row 1) is:

$$\{9098 = on, 9039 = cold\} \longrightarrow \{9061 = comfort, 9096 = cold\}$$

which changing the identifiers of the sensors to a more descriptive name results in:

$$\{Setup\ PAN = on, Output\ temperature = cold\} \longrightarrow$$
$$\{PAN\ temperature = comfort, PAS\ temperature = cold\}$$

In this rule we can observe how the general operation of the building is described, i.e. outside the temperature is cold, the heating setup is ON and for that section of the building (PAN represents north area) the comfort temperature is achieved. In addition we can see that other sections of the building such as PAS (representing the south area) is cold at the same time, so we could determine that there are two rooms or sections that are not usually occupied at the same time.

On the other hand in Figure 21 some rules have been selected, where different behaviours of the building can be seen. For example this rule obtained in winter:

$$\{9047 = off, Humidity = humid, temperature = cold\} \longrightarrow$$
$$\{9039 = comfort, 9096 = on, 9098 = on\}$$

is equivalent to:

$$\{Windows\ PAN = off, Humidity = humid,$$
$$Temperature = cold\} \longrightarrow$$
$$\{Output\ temperature\ PAN = comfort, Setup\ PAS = on,$$
$$Setup\ PAN = on\}$$

This rule gives important information about the relationship between the windows being closed, humidity, a cold day and the heating thermostats on.

The following rule is obtained in summer, as we can see the heating equipment is not working and the windows are open.

$$\{WindowsPAN = on, Windows\ PAS = on\} \longrightarrow$$
$$\{Output\ temperature\ warm, Setup\ PAS = off,$$
$$Setup\ PAN = off, Temperature = comfort\}$$

In addition, we have studied the number of coincident association rules according to the seasons. In Figure 19 we can see the percentage of coinciding rules for every pair of seasons, which are usually those generated by the building system independently of the temperature and weather variables (e.g. windows, lights, security systems...) and those that are different, usually involving temperature and meteorological features.

The following rule occurs equally in summer and winter. It shows us the relationship between the lights on, the tempera-

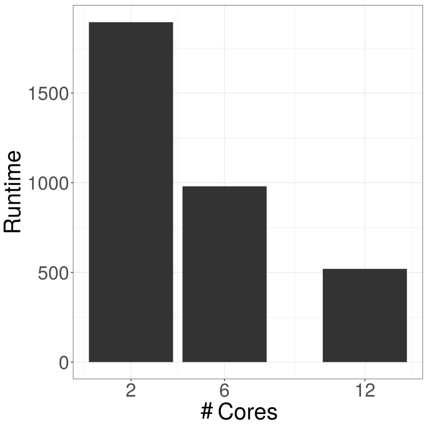


Fig. 14. Time in seconds with different core configurations

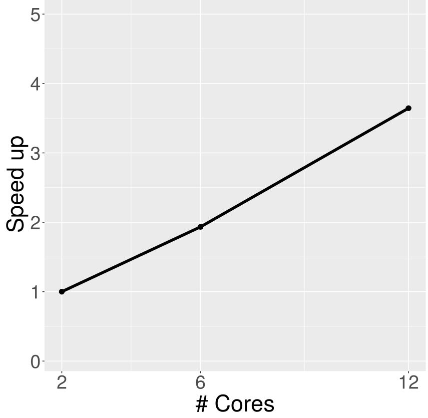


Fig. 15. Speedup versus number of processing cores

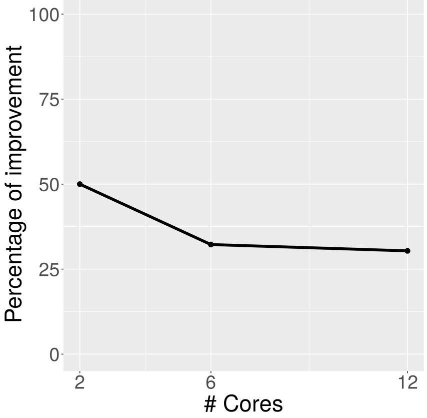


Fig. 16. Efficiency versus number of processing cores

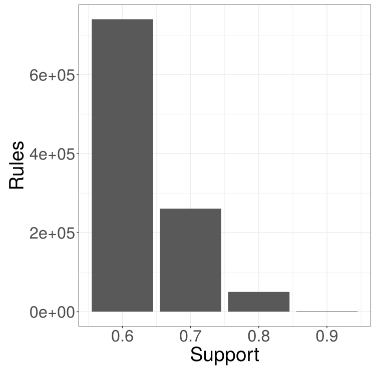


Fig. 17. Number of rules obtained with different parameters of the extraction algorithm

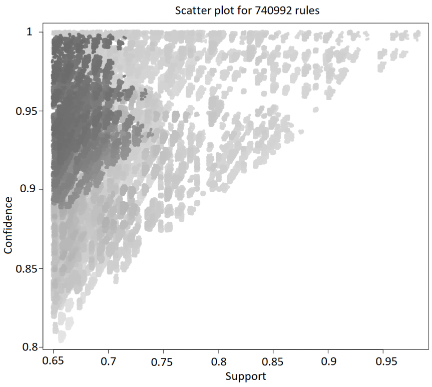


Fig. 18. Rules extracted for the office building in Bucharest according to their support (x-axis) and confidence (y-axis)

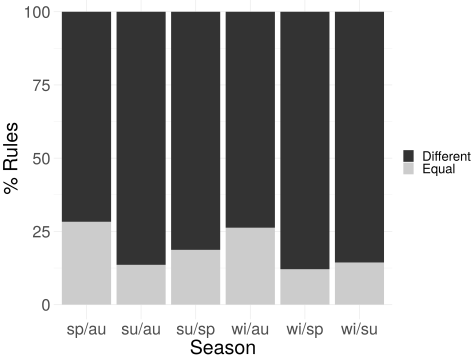


Fig. 19. Comparison of rules obtained by season (sp=spring, au=autumn, su=summer, wi=winter)

ture of the PAN zone in comfort and the high occupation of the building.

$$\{Occupacy : high, \\ Status - Lighting : PAS = on\} \to \\ \{Output\ Status - Lighting : PAN = on, \\ TemperaturePAN = comfort\}$$

## V. Future challenges

For the data pre-processing, we have incorporated two different ways of using the information provided by the user and an automatic method using the data distribution. The latter could be used to suggest this information to the end users. This could be a precursor of a decision making system. Different proposals for this type of systems can be found in [36]. In [37], decision support systems are used to improve the operation of building elements or equipment maintenance. Furthermore, some of them use rules obtained from the behaviour of the users [38] to improve the building functioning. Therefore, using the results provided by our proposal a decision system could be implemented incorporating the real functioning of the building.

Additionally, different methods have been observed in the literature for the management of uncertainty such as the use of polyhedral uncertainty [39]. One of the applications of these techniques is to use RMARS (Robustification of multivariate adaptive regression spline under polyhedral uncertainty) to predict electricity consumption [40], gas consumption [41] or even finances [39]. In future works it can be studied how to combine these types of models with the presented approach, taking benefit of both proposals.

## VI. Conclusions

The discovery and exploitation of information collected from buildings has attracted attention in the last decade due to its economic and environmental impact. Big Data offers a suitable framework for the efficient implementation of analysis techniques capable of handling large amounts of data, especially those produced in building management systems. In addition, the use of fuzzy logic can improve the interpretability of collected sensor data, offering improved results and interpretation to end users.

In this study, a data mining methodology has been implemented using the Big Data framework and applied to different

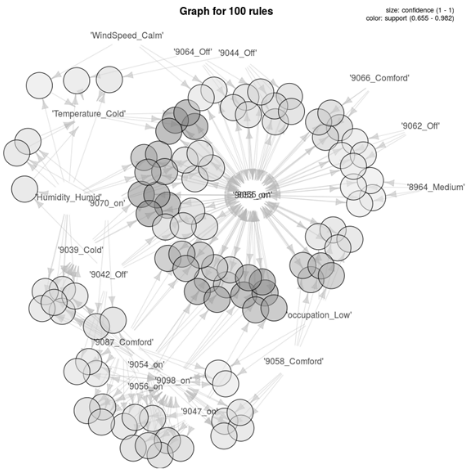


Fig. 20. Graph visualization of some association rules discovered for the office building in Bucharest

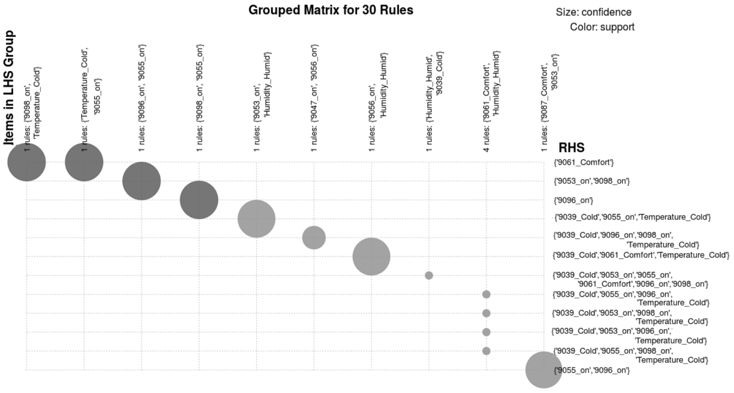


Fig. 21. Some fuzzy association rules discovered for the office building in Bucharest.

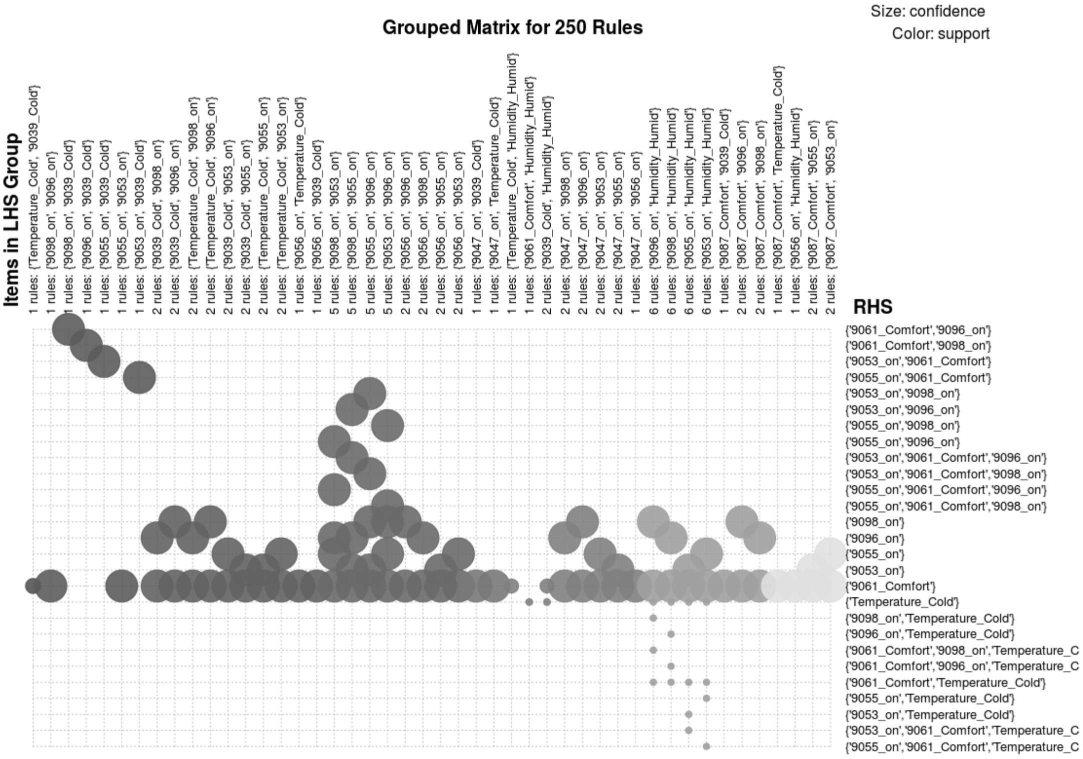


Fig. 22. Some association rules discovered for the office building in Bucharest. LHS stands for Left Hand Side of the rule or Antecedent and RHS for Right Hand Side or consequent.

data sets collected from an office building in Romania. In particular, we have applied a fuzzification algorithm to improve the application of data mining techniques such as association rules. The whole system has been deployed using the Spark platform to enable the analysis of such an amount of data generated by the sensors in the building. This technique has allowed the exploitation of different kinds of data collected in the diverse pilot areas of the building. The proposed solution has been applied to the static data collected from the building, obtaining different relationships that show the energy behaviour of the building and make evident some patterns that can be used to improve the energy efficiency of the building. However, the barrier that arises to automatically analyse continuously generated data needs to be be dealt with. This leads us to propose a future improvement of the proposed system for handling such a continuous flow and to process it in real time conveniently. To do so, there are recently developed utilities within the Spark framework that enables the processing stream data called Spark Streaming [42], [43]. This extension will enable live data streams to be processed by dividing them into batches which can be then processed by the Spark mining algorithms.

Another future improvement concerns the display of results which should be more informative and complete for end users. In fact, there are some applications available [44] that can be conveniently adapted to illustrate the discovered patterns.

## Acknowledgments

The authors gratefully acknowledge the help provided for this research by the owners of the Bucharest Office building in Romania, and the support of the Spanish Ministries of Science, Innovation and Universities (TIN2017-91223-EXP) and Economy and Competitiveness (TIN2015-64776-C3-1-R); and the European Union (Energy IN TIME EeB.NMP.2013-4, No. 608981).

## References

[1] M. S. Kiran, E. Özceylan, M. Gündüz, and T. Paksoy, "Swarm intelligence approaches to estimate electricity energy demand in turkey," *Knowl.-Based Syst.*, vol. 36, pp. 93–103, 2012.

[2] G. Nalcaci, A. Özmen, and G. Weber, "Long-term load forecasting: models based on mars, ANN and LR methods," *CEJOR*, vol. 27, no. 4, pp. 1033–1049, 2019.

[3] F. Yerlikaya-Özkurt, A. Askan, and G. Weber, "A hybrid computational method based on convex optimization for outlier problems: Application to earthquake ground motion prediction," *Informatica, Lith. Acad. Sci.*, vol. 27, no. 4, pp. 893–910, 2016.

[4] S. Midya and S. K. Roy, "Analysis of interval programming in different environments and its application to fixed-charge transportation problem," *Discrete Math., Alg. and Appl.*, vol. 9, no. 3, pp. 1750040:1–1750040:17, 2017.

[5] H. Karau, A. Konwinski, P. Wendell, and M. Zaharia, *Learning Spark: Lightning-Fast Big Data Analysis.* " O'Reilly Media, Inc.", 2015.

[6] Z. Yu, B. C. Fung, and F. Haghighat, "Extracting knowledge from building-related data. a data mining framework," *Building Simulation*, vol. 6, no. 2, pp. 207–222, 2013.

[7] Z. J. Yu, F. Haghighat, and B. C. Fung, "Advances and challenges in building engineering and data mining applications for energy-efficient communities," *Sustainable Cities and Society*, vol. 25, pp. 33–38, 2016.

[8] M. Molina-Solana, M. Ros, M. D. Ruiz, J. Gómez-Romero, and M. Martin-Bautista, "Data science for building energy management: a review," *Renewable and Sustainable Energy Reviews*, vol. 70, pp. 598–609, 2017.

[9] C. Fan and F. Xiao, "Mining gradual patterns in big building operational data for building energy efficiency enhancement," *Energy Procedia*, vol. 143, pp. 119–124, 2017.

[10] T. Ahmad, H. Chen, Y. Guo, and J. Wang, "A comprehensive overview on the data driven and large scale based approaches for forecasting of building energy demand: A review," *Energy and Buildings*, vol. 165, pp. 301–320, 2018.

[11] R. U. Islam, M. S. Hossain, and K. Andersson, "A novel anomaly detection algorithm for sensor data under uncertainty," *Soft Computing*, vol. 22, no. 5, pp. 1623–1639, 2018.

[12] R. Agrawal, T. Imielinski, and A. Swami, "Mining associations between sets of items in large databases," in *ACM-SIGMOD International Conference on Data*, pp. 207–216, 1993.

[13] P. Hájek, "The question of a general concept of the GUHA method," *Kybernetika*, vol. 4, pp. 505–515, 1968.

[14] E. Hüllermeier and Y. Yi, "In defense of fuzzy association analysis," *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, vol. 37, no. 4, pp. 1039–1043, 2007.

[15] J. Calero, G. Delgado, M. Sánchez-Marañón, D. Sánchez, M. A. V. Miranda, and J. Serrano, "An experience in management of imprecise soil databases by means of fuzzy association rules and fuzzy approximate dependencies," in *ICEIS 2004, Proceedings of the 6th International Conference on Enterprise Information Systems, Porto, Portugal, April 14-17, 2004*, pp. 138–146, 2004.

[16] F. Berzal, M. Delgado, D. Sánchez, and M. Vila, "Measuring accuracy and interest of association rules: A new framework," *Intelligent Data Analysis*, vol. 6, no. 3, pp. 221–235, 2002.

[17] M. Delgado, N. Marín, D. Sánchez, and M. Vila, "Fuzzy association rules: General model and applications," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 2, pp. 214–225, 2003.

[18] N. Marín, M. Ruiz, and D. Sánchez, "Fuzzy frameworks for mining data associations: fuzzy association rules and beyond," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 6, pp. 50–69, jan 2016.

[19] M. Delgado, M. Ruiz, D. Sánchez, and J. Serrano, "A formal model for mining fuzzy rules using the RL representation theory," *Information Sciences*, vol. 181, no. 23, pp. 5194–5213, 2011.

[20] M. D. Ruiz, D. Sánchez, M. Delgado, and M. J. Martin-Bautista, "Discovering fuzzy exception and anomalous rules," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 4, pp. 930–944, 2016.

[21] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[22] L. Liu, *Performance comparison by running benchmarks on Hadoop, Spark and Hamr.* PhD thesis, University of Delaware, 2016.

[23] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pp. 2–2, USENIX Association, 2012.

[24] K. Banker, *MongoDB in action.* Manning Publications Co., 2011.

[25] C. Győrödi, R. Győrödi, G. Pecherle, and A. Olah, "A comparative study: Mongodb vs. mysql," in *2015 13th International Conference on Engineering of Modern Electric Systems (EMES)*, pp. 1–6, IEEE, 2015.

[26] E. J. Knibbe, "Building management system," Oct. 15 1996. US Patent 5,565,855.

[27] S. Boschi and G. Santomaggio, *RabbitMQ Cookbook.* Packt Publishing, 2013.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[29] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.

[30] C. Fernandez-Bassso, M. D. Ruiz, and M. J. Martin-Bautista, "Fuzzy association rules mining using spark," in *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp. 15–25, Springer, 2018.

[31] C. Fernandez-Basso, M. Ruiz, and M. Martin-Bautista, "Some proposals for fuzzy association rules mining using spark," *Submitted to Research in Big Data*, 2019.

[32] V. P. Kumar and A. Gupta, "Analyzing scalability of parallel algorithms and architectures," *Journal of parallel and distributed computing*, vol. 22, no. 3, pp. 379–391, 1994.

[33] A. Y. Grama, A. Gupta, and V. Kumar, "Isoefficiency: Measuring the scalability of parallel algorithms and architectures," *IEEE Parallel & Distributed Technology: Systems & Applications*, vol. 1, no. 3, pp. 12–21, 1993.

[34] C. Barba-González, J. García-Nieto, A. Benítez-Hidalgo, A. J. Nebro, and J. F. Aldana-Montes, "Scalable inference of gene regulatory networks with the spark distributed computing platform," in *Intelligent Distributed Computing XII* (J. Del Ser, E. Osaba, M. N. Bilbao, J. J. Sanchez-Medina, M. Vecchio, and X.-S. Yang, eds.), (Cham), pp. 61–70, Springer International Publishing, 2018.

[35] F. J. Baldán and J. M. Benítez, "Distributed fastshapelet transform: a big data time series classification algorithm," *Information Sciences*, 2018.

[36] F. Frombo, R. Minciardi, M. Robba, and R. Sacile, "A decision support system for planning biomass-based energy production," *Energy*, vol. 34, no. 3, pp. 362–369, 2009.

[37] A. Mattiussi, M. Rosano, and P. Simeoni, "A decision support system for sustainable energy supply combining multi-objective and multi-attribute analysis: An australian case study," *Decision Support Systems*, vol. 57, pp. 150–159, 2014.

[38] Y.-K. Juan, P. Gao, and J. Wang, "A hybrid decision support system for sustainable office building renovation and energy performance improvement," *Energy and buildings*, vol. 42, no. 3, pp. 290–297, 2010.

[39] A. Özmen and G. W. Weber, "Rmars: robustification of multivariate adaptive regression spline under polyhedral uncertainty," *Journal of Computational and Applied Mathematics*, vol. 259, pp. 914–924, 2014.

[40] M. H. Yıldırım, A. Özmen, Ö. T. Bayrak, and G. W. Weber, "Electricity price modelling for turkey," in *Operations Research Proceedings 2011*, pp. 39–44, Springer, 2012.

[41] G.-W. Weber, A. zmen, and Y. Zinchenko, *RMARS under Cross-Polytope Uncertainty - Prediction of Natural Gas Consumption*, p. 25. 10 2019.

[42] C. Prakash, "Spark streaming vs flink vs storm vs kafka streams vs samza : Choose your stream processing framework," *Medium*, 2018. Accessed 06-02-2019 https://medium.com/@chandanbaranwal/spark-streaming-vs-flink-vs-storm-vs-kafka-streams-vs-samza-choose-your-stream-processing-91ea3f04675b.

[43] C. Fernandez-Basso, A. J. Francisco-Agra, M. J. Martin-Bautista, and M. D. Ruiz, "Finding tendencies in streaming data using big data frequent itemset mining," *Knowledge-Based Systems*, vol. 163, pp. 666–674, 2019.

[44] C. Fernandez-Basso, M. D. Ruiz, M. Delgado, and M. J. Martin-Bautista, "A comparative analysis of tools for visualizing association rules: A proposal for visualising fuzzy association rules," in *2019 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (EUSFLAT 2019)*, Atlantis Press, 2019.

**Carlos Fernandez-Basso** received the degree in computer science and the M.Sc. degree in data science from the Universidad de Granada, in 2014 and 2015, respectively, where he is currently pursuing the Ph.D. degree in computer science and energy efficiency. He was a Lead Developer in the EU FP7 Project Energy IN TIME in the topics of building simulation and control, data analytics, and machine learning. He also collaborates with the Data Science Institute, Imperial College London, where he has carried out research stays, from 2016 to 2018.

**M.Dolores Ruiz** Dr. M. Dolores Ruiz got the Mathematics degree in 2005 and the European PhD. in Computer Science in 2010, both from the University of Granada. She held non-permanent teaching positions in the Universities of Jaén, Granada and Cádiz. She is currently working at the Statistics and Operative Research department in the University of Granada and she belongs to the Approximate Reasoning and AI research group. She has organized several special sessions about Data Mining in International conferences and is part of the organization committee of the FQAS'2013 and SUM'2017 conferences. Her research areas of interest are: data mining, information retrieval, correlation statistical measures, sentence quantification and fuzzy sets theory. Her expertise involves knowledge extraction from databases involving uncertainty using association rules, exception rules, anomalous rules and gradual dependences, as well as, formal modelling for the representation and evaluation of association rules.

**Maria J. Martin-Bautista** Dr. Maria J. Martin-Bautista is a Full Professor at the Department of Computer Science and Artificial Intelligence at the University of Granada, Spain, since 1997. She is a member of the IDBIS (Intelligent Data Bases and Information Systems) research group. Her current research interests include Big Data Analytics in Data, Text and Web Mining, Intelligent Information Systems, Knowledge Representation and Uncertainty. She has supervised several Ph. D. Thesis and published more than 100 papers in high impact international journals and conferences. She has participated in more than 20 R+D projects and has supervised several research technology transfers with companies. She has served as a program committee member for several international conferences.

# Bibliography

[AIS93a] Agrawal R., Imielinski T., and Swami A. (1993) Mining associations between sets of items in large databases. In *ACM-SIGMOD Int. Conf. on Data*, pp. 207–216.

[AIS93b] Agrawal R., Imieliński T., and Swami A. (1993) Mining association rules between sets of items in large databases. *SIGMOD Rec.* 22(2): 207–216.

[AJS14] Afram A. and Janabi-Sharifi F. (2014) Theory and applications of hvac control systems–a review of model predictive control (mpc). *Building and Environment* 72: 343–355.

[Alp20] Alpaydin E. (2020) *Introduction to machine learning.* MIT press.

[AMS$^+$96] Agrawal R., Mannila H., Srikant R., Toivonen H., Verkamo A. I., *et al.* (1996) Fast discovery of association rules. *Advances in knowledge discovery and data mining* 12(1): 307–328.

[AS$^+$94] Agrawal R., Srikant R., *et al.* (1994) Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volumen 1215, pp. 487–499.

[BBSV01] Berzal F., Blanco I., Sánchez D., and Vila M.-A. (2001) A new framework to assess association rules. In *Advances in Intelligent Data Analysis*, pp. 95–104. Springer.

[BCSV07] Berzal F., Cubero J., Sánchez D., and Vila M. (2007) An alternative approach to discover gradual dependencies. *Int. Journal of Uncertainty, Fuzziness and Knowledge-based Systems* 15(5): 559–570.

[BD08] Bruzzese D. and Davino C. (2008) Visual mining of association rules. In *Visual Data Mining*, pp. 103–122. Springer.

[BDSV02] Berzal F., Delgado M., Sánchez D., and Vila M. (2002) Measuring accuracy and interest of association rules: A new framework. *Intelligent Data Analysis* 6(3): 221–235.

[BHJ09] Bastian M., Heymann S., and Jacomy M. (2009) Gephi: An open source software for exploring and manipulating networks. In *Third international AAAI conference on weblogs and social media.*

[BJA99] Bayardo Jr R. J. and Agrawal R. (1999) Mining the most interesting rules. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 145–154. Citeseer.

[BOH11] Bostock M., Ogievetsky V., and Heer J. (2011) D$^3$ data-driven documents. *IEEE transactions on visualization and computer graphics* 17(12): 2301–2309.

[Bok14] Bokeh Development Team (2014) Bokeh: Python library for interactive visualization. `http://www.bokeh.pydata.org`, Last accessed on 2019-05-30.

[Bre01] Breiman L. (2001) Random forests. *Machine learning* 45(1): 5–32.

[CDS$^+$04] Calero J., Delgado G., Sánchez-Marañón M., Sánchez D., Miranda M. A. V., and Serrano J. (2004) An experience in management of imprecise soil databases by means of fuzzy association rules and fuzzy approximate dependencies. In *ICEIS 2004, Porto, Portugal, April 14-17, 2004*, pp. 138–146.

[CL07] Chen J. and Li. S. (2007) *Gc-tree: a fast online algorithm for mining frequent closed itemsets*, pp. 457–468.

[CM07] Cherkassky V. and Mulier F. M. (2007) *Learning from data: concepts, theory, and methods.* John Wiley & Sons.

[CSXD12] Chen H., Shu L., Xia J., and Deng Q. (2012) Mining frequent pattern in varying-size sliding window of online transactional data streams. *Information Sciences* 215: 15–36.

[CXSP17] Chen W., Xie C., Shang P., and Peng Q. (2017) Visual analysis of user-driven association rule mining. *Journal of Visual Languages & Computing* 42: 76–85.

[DC19] Dattolo A. and Corbatto M. (2019) Visualbib: A novel web app for supporting researchers in the creation, visualization and sharing of bibliographies. *Knowledge-Based Systems* 182: 104860.

[DHP06] Dubois D., Hüllermeier E., and Prade H. (Sep 2006) A systematic approach to the assessment of fuzzy association rules. *Data Mining and Knowledge Discovery* 13(2): 167–192.

[DHS12] Duda R. O., Hart P. E., and Stork D. G. (2012) *Pattern classification.* John Wiley & Sons.

[DMSV03] Delgado M., Marín N., Sánchez D., and Vila M. (2003) Fuzzy association rules: General model and applications. *IEEE Transactions on Fuzzy Systems* 11(2): 214–225.

[DRS11] Delgado M., Ruiz M., and Sánchez D. (2011) New approaches for discovering exception and anomalous rules. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 19(2): 361–399.

[DRSS11a] Delgado M., Ruiz M., Sánchez D., and Serrano J. (2011) A formal model for mining fuzzy rules using the RL representation theory. *Information Sciences* 181(23): 5194–5213.

[DRSS11b] Delgado M., Ruiz M. D., Sánchez D., and Serrano J.-M. (2011) A formal model for mining fuzzy rules using the rl representation theory. *Information Sciences* 181(23): 5194–5213.

[FBFAMBR19] Fernandez-Basso C., Francisco-Agra A. J., Martin-Bautista M. J., and Ruiz M. D. (2019) Finding tendencies in streaming data using big data frequent itemset mining. *Knowledge-Based Systems* 163: 666–674.

[FBRMB16] Fernandez-Basso C., Ruiz M. D., and Martin-Bautista M. J. (2016) Extraction of association rules using big data technologies. *International Journal of Design & Nature and Ecodynamics* 11(3): 178–185.

[GBOZ18] Gil S., Bobadilla J., Ortega F., and Zhu B. (2018) Visualrs: Java framework for visualization of recommender systems information. *Knowledge-Based Systems* 155: 66–70.

[GCC05] Gabroveanu M., Cosulschi M., and Constantinescu N. (2005) A new approach to mining fuzzy association rules from distributed databases. *Annals of the University of Bucharest* LIV: 3–16.

[GCS16] Gabroveanu M., Cosulschi M., and Slabu F. (2016) Mining fuzzy association rules using mapreduce technique. In *Int. Symposium on INnovations in Intelligent SysTems and Applications*, INISTA, pp. 1–8.

[GICC07] Gabroveanu M., Iancu I., Cosulschi M., and Constantinescu N. (2007) Towards using grid services for mining fuzzy association rules. In *Proc. of the 1st East European Workshop on Rule-Based Applications*, RuleApps, pp. 507–513.

[HÖ2] Hüllermeier E. (2002) Association rules for expressing gradual dependencies. In *Proc. PKDD 2002 Lecture Notes in Computer Science, 2431*, pp. 200–211.

[Hah17] Hahsler M. (2017) ArulesViz: interactive visualization of association rules with R. *The R Journal* 9(2): 1.

[Har75] Hartigan J. A. (1975) *Clustering algorithms.* Wiley series in probability and mathematical statistics. Wiley, New York, NY.

[HC11] Hahsler M. and Chelluboina S. (2011) Visualizing association rules: Introduction to the R-extension package ArulesViz. *R project module* pp. 223–238.

[HPY00] Han J., Pei J., and Yin Y. (2000) Mining frequent patterns without candidate generation. *ACM Sigmod Record* 29(2): 1–12.

[HSSC08] Hagberg A., Swart P., and S. Chult D. (2008) Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

[HSW00] Hofmann H., Siebes A. P. J. M., and Wilhelm A. F. X. (2000) Visualizing association rules with interactive mosaic plots. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pp. 227–235. ACM, New York, NY, USA.

[HY07] Hüllermeier E. and Yi Y. (2007) In defense of fuzzy association analysis. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics* 37(4): 1039–1043.

[IES] IESiesve software), howpublished = `https://www.iesve.com/software`, note = Accessed: 2020-04-30.

[JGW10] Juan Y.-K., Gao P., and Wang J. (2010) A hybrid decision support system for sustainable office building renovation and energy performance improvement. *Energy and buildings* 42(3): 290–297.

[JPA08] Jorge A., Poças J., and Azevedo P. J. (2008) A methodology for exploring association models. In *Visual Data Mining*, pp. 46–59. Springer.

[KD07] Koh J. and Don Y. (2007) Approximately mining recently representative patterns on data streams. In *Proceedings of PAKDD Pacific-Asia Conference on Knowledge Discovery and Data Mining*, volumen 4819, pp. 231–243.

[KKC12] Kim H. K., Kim J. K., and Chen Q. Y. (2012) A product network analysis for extending the market basket analysis. *Expert Systems with Applications* 39(8): 7403–7410.

[KKWZ15] Karau H., Konwinski A., Wendell P., and Zaharia M. (2015) *Learning Spark: Lightning-Fast Big Data Analysis.* " O'Reilly Media, Inc.".

[KMCG17] Koochaksaraei R. H., Meneghini I. R., Coelho V. N., and Guimarães F. G. (2017) A new visualization method in many-objective optimization with chord diagram and angular mapping. *Knowledge-Based Systems* 138: 134–154.

[KÖGP12] Kiran M. S., Özceylan E., Gündüz M., and Paksoy T. (2012) Swarm intelligence approaches to estimate electricity energy demand in turkey. *Knowl.-Based Syst.* 36: 93–103.

[KT03] Kopanakis I. and Theodoulidis B. (2003) Visual data mining modeling techniques for the visualization of mining outcomes. *Journal of Visual Languages & Computing* 14(6): 543–589.

[LC10] Leung C. K.-S. and Carmichael C. L. (2010) FpVAT: a visual analytic tool for supporting frequent pattern mining. *ACM SIGKDD Explorations Newsletter* 11(2): 39–48.

[LHL09] Li H., Ho C., and Lee S. (2009) Incremental updates of closed frequent itemsets over continuous data streams. *Expert Systems with Applications* 36(2): 2451–2458.

[Li15] Li K. (2015) On integrating information visualization techniques into data mining: A review. *arXiv preprint arXiv:1503.00202* .

[LIC08] Leung C. K.-S., Irani P. P., and Carmichael C. L. (2008) WiFIsViz: effective visualization of frequent itemsets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pp. 875–880. Citeseer.

[LL09] Li H. and Lee S. (2009) Mining frequent itemsets over data streams using efficient window sliding techniques. *Expert Systems with Applications* 36(2): 1466–1477.

[LPPM07] Lopes A. A., Pinho R., Paulovich F. V., and Minghim R. (2007) Visual text mining using association rules. *Computers & Graphics* 31(3): 316–326.

[LWZ$^+$08] Li H., Wang Y., Zhang D., Zhang M., and Chang E. Y. (2008) PFP: parallel fp-growth for query recommendation. In *Proceedings of the 2008 ACM conference on Recommender systems*, pp. 107–114. ACM.

[LY00] Louie E. and Young T. (2000) Finding association rules using fast bit computation: Machine-oriented modeling. In *International Symposium on Methodologies for Intelligent Systems*, pp. 486–494. Springer.

[LZ11] Li H. and Zhang N. (2011) *A false negative maximal frequent itemset mining algorithm over stream*, pp. 29–41.

[LZC12] Li H., Zhang N., and Chen Z. (2012) A simple but effective maximal frequent itemset mining algorithm over streams. *Journal of Software* 7(1): 25–32.

[LZZ$^+$14] Li H., Zhang N., Zhu J., Cao H., and Wang Y. (2014) Efficient frequent itemset mining methods over time-sensitive streams. *Knowledge-Based Systems* 56: 281 – 298.

[MBGCT16] Martínez V., Berzal Galiano F., and Cubero Talavera J. C. (2016) The NOESIS network-oriented exploration, simulation, and induction system. *CoRR* abs/1611.04810.

[MBY$^+$16] Meng X., Bradley J., Yavuz B., Sparks E., Venkataraman S., Liu D., Freeman J., Tsai D., Amde M., Owen S., *et al.* (2016) Mllib: Machine learning in apache spark. *The Journal of Machine Learning Research* 17(1): 1235–1241.

[MD16] Mirakhorli A. and Dong B. (2016) Occupancy behavior based model predictive control for building indoor climate—a critical review. *Energy and Buildings* 129: 499–513.

[MFMSG20] Martínez V., Fernando S., Molina-Solana M., and Guo Y. (2020) Tuoris: A middleware for visualizing dynamic graphics in scalable resolution display environments. *Future Generation Computer Systems* 106: 559–571.

[MR17] Midya S. and Roy S. K. (2017) Analysis of interval programming in different environments and its application to fixed-charge transportation problem. *Discrete Math., Alg. and Appl.* 9(3): 1750040:1–1750040:17.

[MRS14] Mattiussi A., Rosano M., and Simeoni P. (2014) A decision support system for sustainable energy supply combining multi-objective and multi-attribute analysis: An australian case study. *Decision Support Systems* 57: 150–159.

[MRS16] Marín N., Ruiz M., and Sánchez D. (jan 2016) Fuzzy frameworks for mining data associations: fuzzy association rules and beyond. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 6(2): 50–69.

[NÖW19] Nalcaci G., Özmen A., and Weber G. (2019) Long-term load forecasting: models based on mars, ANN and LR methods. *CEJOR* 27(4): 1033–1049.

[OONL02] ONG H.-H., Ong K.-L., Ng W.-K., and LIM E. P. (2002) Crystalclear: Active visualization of association rules. In *ICDM'02 International Workshop on Active Mining AM2002*.

[PHMA$^+$04] Pei J., Han J., Mortazavi-Asl B., Wang J., Pinto H., Chen Q., Dayal U., and Hsu M.-C. (2004) Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on knowledge and data engineering* 16(11): 1424–1440.

[Pra18] Prakash C. (2018) Spark streaming vs flink vs storm vs kafka streams vs samza : Choose your stream processing framework. *Medium* Accessed 06-02-2019 https://medium.com/@chandanbaranwal/spark-streaming-vs-flink-vs-storm-vs-kafka-streams-vs-samza-choose-your-stream-processing-91ea3f04675b.

[QGYH14] Qiu H., Gu R., Yuan C., and Huang Y. (2014) Yafim: A parallel frequent itemset mining algorithm with spark. In *Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International*, pp. 1664–1671. IEEE.

[RKK15] Rathee S., Kaul M., and Kashyap A. (2015) R-apriori: An efficient apriori based algorithm on spark. In *Proceedings of the PIKM'15*, pp. 27–34. ACM, Melbourne, VIC, Australia.

[RŠ05] Rauch J. and Šimůnek M. (2005) *Foundations of Data Mining and knowledge Discovery*, chapter An Alternative Approach to Mining Association Rules, pp. 211–231. Springer Berlin Heidelberg, Berlin, Heidelberg.

[RSDMB16] Ruiz M. D., Sánchez D., Delgado M., and Martin-Bautista M. J. (2016) Discovering fuzzy exception and anomalous rules. *IEEE Transactions on Fuzzy Systems* 24(4): 930–944.

[SG08] Simoff S. J. and Galloway J. (2008) Visual discovery of network patterns of interaction between attributes. In *Visual Data Mining*, pp. 172–195. Springer.

[SGM15] Singh S., Garg R., and Mishra P. (2015) Performance analysis of apriori algorithm with different data structures on hadoop cluster. *International Journal of Computer Applications* 128(9): 45–51.

[SPH$^+$17] Sievert C., Parmer C., Hocking T., Chamberlain S., Ram K., Corvellec M., and Despouy P. (2017) Plotly: Create Interactive Web Graphics. *R package version* 4(1): 1.

[SS13] Sagiroglu S. and Sinanc D. (2013) Big data: A review. In *2013 international conference on collaboration technologies and systems (CTS)*, pp. 42–47. IEEE.

[SSPB16] Simard F., St-Pierre J., and Biskri I. (2016) Mining and visualizing robust maximal association rules on highly variable textual data in entrepreneurship. In *Proceedings of the 8th International Conference on Management of Digital EcoSystems*, pp. 215–222. ACM.

[TAJL09] Tanbeer S., Ahmed C., Jeong B., and Lee Y. (2009) Sliding window-based frequent pattern mining over data streams. *Information sciences* 179(22): 3843–3865.

[TBH17] Tyner S., Briatte F., and Hofmann H. (2017) Network visualization with ggplot2. *The R Journal* .

[UHB01] Unwin A., Hofmann H., and Bernt K. (2001) The twokey plot for multiple association rules control. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 472–483. Springer.

[VRM18] Valle M. A., Ruz G. A., and Morrás R. (2018) Market basket analysis: Complementing association rules with minimum spanning trees. *Expert Systems with Applications* 97: 146–162.

[Whi12] White T. (2012) *Hadoop: The definitive guide.* O'Reilly Media, Inc.

[WWT99] Wong P. C., Whitney P., and Thomas J. (1999) Visualizing association rules for text mining. In *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis' 99)*, pp. 120–123. IEEE.

[WZWD13] Wu X., Zhu X., Wu G.-Q., and Ding W. (2013) Data mining with big data. *IEEE transactions on knowledge and data engineering* 26(1): 97–107.

[XC05] Xu J. and Chen H. (Junio 2005) Criminal network analysis and visualization. *Commun. ACM* 48(6): 100–107.

[Yan03] Yang L. (2003) Visualizing frequent itemsets, association rules, and sequential patterns in parallel coordinates. In *International Conference on Computational Science and Its Applications*, pp. 21–30. Springer.

[Yan05] Yang L. (2005) Pruning and visualizing generalized association rules in parallel coordinates. *IEEE Transactions on Knowledge and Data Engineering* 17(1): 60–70.

[Yan08] Yang L. (2008) Visual exploration of frequent itemsets and association rules. In *Visual Data Mining*, pp. 60–75. Springer.

[YAW16] Yerlikaya-Özkurt F., Askan A., and Weber G. (2016) A hybrid computational method based on convex optimization for outlier problems: Application to earthquake ground motion prediction. *Informatica, Lith. Acad. Sci.* 27(4): 893–910.

[Zak00] Zaki M. J. (2000) Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering* 12(3): 372–390.

[ZCD$^+$12a] Zaharia M., Chowdhury M., Das T., Dave A., Ma J., McCauley M., Franklin M., Shenker S., and Stoica I. (2012) Discretized streams: Fault-tolerant streaming computation at scale. In *ACM Symposium on Operating Systems Principles (SOSP'13).*

[ZCD$^+$12b] Zaharia M., Chowdhury M., Das T., Dave A., Ma J., McCauley M., Franklin M. J., Shenker S., and Stoica I. (2012) Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proc. of the 9th USENIX conference on Networked Systems Design and Implementation*, pp. 2–2. USENIX Association.

[ZDLH13] Zaharia M., Das T., Li H., and Hunter T. (2013) Discretized streams: Fault-tolerant streaming computation at scale. In *ACM Symposium on Operating Systems Principles (SOSP'13)*, pp. 423–438.