Universidad de Granada

Departamento de Ciencias de la Computacin
e Inteligencia Artificial

Dataset Shift in Classification: Terminology, Benchmarks and Methods

Tesis Doctoral

Jose García Moreno-Torres

Granada, Enero de 2013

Editor: Editorial de la Universidad de Granada
Autor: José García Moreno Torres
D.L.: GR 1896-2013
ISBN: 978-84-9028-588-6

Universidad de Granada

Dataset Shift in Classification: Terminology, Benchmarks and Methods

MEMORIA QUE PRESENTA

Jose García Moreno-Torres

PARA OPTAR AL GRADO DE DOCTOR EN INFORMÁTICA

Enero de 2013

DIRECTOR

Dr. Francisco Herrera Triguero

Departamento de Ciencias de la Computacin
e Inteligencia Artificial

La memoria titulada "*Dataset Shift in Classification: Terminology, Benchmarks and Methods*", que presenta D. Jose García Moreno-Torres para optar al grado de doctor, ha sido realizada dentro del Máster Oficial de Doctorado "*Soft Computing y Sistemas Inteligentes*" del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada bajo la dirección del doctor D. Francisco Herrera Triguero.

El doctorando y el director de la tesis de la tesis garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección del director de la tesis, y hasta donde nuestro conocimiento alcanza, en la realización del trabajo se han respetado los derechos de otros autores a ser citados cuando se han utilizado sus resultados o publicaciones.

Granada, Enero de 2013

El Doctorando

Fdo: Jose García Moreno-Torres

El Director

Fdo: Francisco Herrera Triguero

Esta tesis doctoral ha sido desarrollada bajo la financiación de los fondos asociados al proyecto P10-TIC-6858 de la Junta de Andalucía y los proyectos TIN2008-06681-C06-01 y TIN2011-28488 del Ministerio de Ciencia e Innovación. También ha sido subvencionada bajo el programa de becas de Formación de Profesorado Universitario, en la Resolución del 8 de julio de 2009, bajo la referencia AP2008-01594.

Agradecimientos

Esta memoria de tesis está dedicada a todos aquellos cuyo apoyo ha hecho posible su conclusión, espero, satisfactoria.

En primer lugar, en el plano personal, dar las gracias a Rita, que es quien más ha sufrido la dedicación que exige un proyecto como éste; a mis padres Jose Vicente y Rosario, que me iniciaron en el interés por la investigación; a mis hermanos Carlos y Laura, y al resto de familiares y amigos que me han apoyado a lo largo de mi carrera.

En el ámbito profesional, me gustaría comenzar dando las gracias a mi director de tesis, Francisco Herrera, que ha sabido en todo momento encontrar el equilibrio justo entre guiarme y darme la libertad de explorar mis ideas.

Quiero agradecer por último a mis compañeros, tanto del grupo SCI2S como del centro, CITIC, así como a los investigadores con los que he colaborado en mis estancias externas, tanto en Illinois, como en Notre Dame, e incluso en la misma UGR con el grupo EC3. Me gustaría mencionar especialmente a Xavier Llorà y a David Goldberg, con los que me inicié en la investigación, pero espero me permitan omitir el resto de los nombres porque la enumeración sería interminable y me acabaría dejando a alguien de entre todos aquellos con quienes he compartido reflexiones, dudas, ideas más y menos descabelladas; y que se han hecho imprescindibles para la culminación de este trabajo.

GRACIAS A TODOS

Table of Contents

Figure index

Part I. PhD dissertation

1. Introducción

La Minería de Datos es una disciplina de análisis de datos que forma parte del campo más amplio de la Inteligencia Artificial. Su objetivo principal es el de llevar a cabo análisis inteligente de datos a través de la inferencia automática de patrones en los datos, con la intención de construir un modelo capaz de explicar dichos datos, habitualmente prediciendo una o varias variables objetivo. El procedimiento habitual de Minería de Datos se compone de tres fases:

1. **Preprocesamiento de los datos**, en la que se eliminan ejemplos redundantes o ruido, se imputan valores perdidos, corrigen desequilibrios entre clases, etc.

2. **Construcción del modelo**, en la que se aplica un algoritmo de aprendizaje para construir un modelo basado en los datos de entrenamiento.

3. **Validación del modelo**, en la que se presenta un conjunto de prueba compuesto de ejemplos no vistos anteriormente al modelo construido en la etapa anterior, para estimar cómo se comportará cuando se integre en un nuevo entorno.

En función de la información de que disponga el algoritmo de aprendizaje para construir el modelo, los problemas de Minería de Datos pueden dividirse en

- **Aprendizaje supervisado**, donde los valores de la(s) variable(s) objetivo son conocidos para un subconjunto de los datos (el conjunto de entrenamiento).

- **Aprendizaje no supervisado**, donde los valores de la(s) variable(s) objetivo son también desconocidos en el conjunto de entrenamiento.

Este trabajo se centra exclusivamente en aprendizaje supervisado.

Más allá, en función del tipo de variable objetivo del problema, las tareas de aprendizaje supervisado se pueden dividir en

- **Regresión**, donde la(s) variable(s) objetivo es (son) continua(s), es decir, número(s) reales.

- **Clasificación**, donde la(s) variable(s) objetivo es (son) discreta(s), y puede(n) tomar un número finito de valores (etiquetas).

El enfoque de este trabajo se centra en problemas de Clasificación con una única variable objetivo y con conjuntos de entrenamiento y prueba estáticos. Por estáticos queremos decir que son completamente conocidos en un momento dado, en contraste con problemas de series temporales en los que los datos están disponibles paso a paso.

Una suposición sobre la que el estudio de este tipo de problemas de Clasificación se ha basado tipicamente es que la distribución $P(y, x)$ es la misma tanto para los datos de entrenamiento como de prueba. Bajo esta suposición, un modelo construido con los datos de entrenamiento y que se ajusta a ellos perfectamente debería predecir las etiquetas de los datos de prueba muy acertadamente.

Sin embargo, hay situaciones, que se dan con frecuencia en aplicaciones reales, en las que la suposición previa no se cumple. Esta cuestión ha sido llamada "Fractura de Datos", y es el principal objeto de estudio de este trabajo. El fenómeno ha sido estudiado en profundidad en análisis de series temporales, pero es relativamente nuevo para clasificación, con la mayoría de los trabajos relevantes publicados en los últimos 5-10 años.

La Fractura de Datos se puede considerar un problema de calidad de los datos, y en está por tanto relacionado con ruido, valores perdidos, análisis de complejidad de datos o no balanceo. Sin embargo, se diferencia de ellos en que no es observable sólo a partir de los datos de entrenamiento, sino que se define como un problema entre los datos de entrenamiento y los de aplicación. Por esta razón, las propuestas para analizar y resolver la Fractura de Datos generalmente no se centran en la fase de preprocesamiento, sino en la adaptación del modelo construido. En este sentido, hay una relación cercana entre los campos de Fractura de Datos y de transferencia de aprendizaje.

En esta memoria de tesis, presentamos la investigación realizada en Fractura de Datos en Clasificación. Comenzamos proponiendo un estándar para la unificación de la terminología asociada al problema, ya que era habitual en la dispersa literatura encontrar el mismo concepto definido con distintos términos, o distintos conceptos asociados al mismo término. Seguidamente creamos una serie de conjuntos de datos de referencia para que sirvan de base para la realización de comparaciones justas entre el comportamiento de las diversas propuestas de la literatura, y después presentamos nuestra propia alternativa. Finalmente, estudiamos cómo interactúa la Fractura de Datos con otros factores en Clasificación como el no balanceo o la validación cruzada con k-subgrupos. Para llevar a cabo estas tareas, hemos estructurado esta memoria en dos partes:

- La parte I está dedicada a la especificación del problema, la discusión de los ángulos específicos empleados para aproximarlo, así como las conclusiones aprendidas.
- La parte II contiene las publicaciones asociadas a este estudio.

En la parte I, tras esta introducción, continuamos con los preliminares que sirven de sustento a este trabajo, mostrados en la Sección 2. Los problemas abiertos que justifican esta tesis están en la Sección 3. Los objetivos que persigue este trabajo se presentan en la Sección 4. En la Sección 5 resumimos los resultados más interesantes obtenidos en los trabajos que componen esta tesis. Finalmente, presentamos las conclusiones globales en la Sección 6, y terminamos con una discussión sobre futuras líneas de investigación que permanecen abiertas en el campo de la Fractura de Datos, en la Sección 7.

En la parte II, incluimos un compendio de cinco publicaciones que desarrollan los objetivos presentados, distribuidas en tres secciones:

- Unificación del estudio de la Fractura de Datos: Revisiones terminológica y experimental.
- Una propuesta para resolver la Fractura de Datos a través de la Extracción de Características basada en Programación Genética.

- Interacciones entre Fractura de Datos y otras cuestiones en clasificación: No balanceo y validación cruzada.

1. Introduction

Data Mining is a data analysis discipline that is part of the broader field of Artificial Intelligence. Its main goal is to perform intelligent analysis of data through the automatic inference of patterns within the data in order to build a model capable of explaining said data, usually predicting one or several target variables. The typical Data Mining procedure is composed of three phases:

1. **Data preprocessing**, where redundant or noisy examples or variables are purged, missing values imputed, class imbalances corrected, etc.
2. **Model building**, where a learning algorithm is applied to construct a model based on the given training data.
3. **Model validation**, where the model built in the previous step is presented a test set made of previously unseen examples, to estimate its performance when deployed in a new environment.

Depending on the information the learning algorithm has available to construct the model, Data Mining problems can be split into

- **Supervised learning**, where the values of the target variable(s) are known for a subset of the data (the training set).
- **Unsupervised learning**, where the values of the target variable(s) are unknown also for the training set. A typical example would be a clustering problem.

This work focuses solely on supervised learning.

Further, depending on the type of target variable of the problem, supervised learning tasks can be divided into

- **Regression**, where the target variable(s) is (are) real-coded, continuous number(s).
- **Classification**, where the target variable(s) is (are) discrete, and a finite number of values (labels) are available.

The scope of this work is centered around Classification problems with a single target variable and with static training and test sets. By static we mean they are fully known at the same time, unlike time series problems where data measurements are made available step by step.

A basic assumption upon which the study of these types of Classification problems has typically relied is that the joint distribution $P(y, x)$ is the same for both the training and test sets. Under this assumption, a model built on the training data and which fits it perfectly should be expect to predict labels very accurately on the test data.

However, there are some cases, which occur often in real world applications, where the above assumption is not true. This issue has been named "Dataset Shift", and is the main object of study of this work. This phenomenon has been studied quite deeply in time series analysis, but is relatively new in Classification, with most relevant publications having appeared in the past 5-10 years.

Dataset Shift can be seen as a data quality problem, and as such is related to noise, missing values, data complexity or imbalance. However, it is different from them in that it is not observable from training data only, but is defined as an issue between the training and test datasets. For

this reason, the approaches to analyze and solve Dataset Shift are typically not focused on the preprocessing part, but rather in the adaptation of the model built. In this regard, there is a close relationship between the fields of Dataset Shift and transfer learning.

In this thesis memory, we present the research performed on Dataset Shift in Classification. We begin by proposing a standard to unify the terminology associated to the problem, since it was a common occurrence in the scattered literature to have the same concept be defined with different terms, or different concepts addressed with the same term. We then create a benchmark of datasets to serve as the basis of fair comparisons among the performances of the different proposals in the literature, and then present our own original solution. Lastly, we study how Dataset Shift interacts with other factors in Classification such as imbalanced datasets or k-fold cross-validation. To perform these tasks, we have structured this memory in two parts:

- Part I is dedicated to the problem statement, the discussion of the specific angles taken to approach it, and the conclusions drawn.
- Part II contains the publications associated with this study.

In Part I, after this Introduction, we continue with some preliminaries this work builds upon, shown in Section 2. The open problems that justify this thesis are in Section 3. The objectives this work pursues are presented in Section 4. In Section 5 we summarize the most interesting results obtained in the studies that comprise this thesis. Finally, we present the overall conclusions of this thesis in Section 6, and end with a discussion of future research avenues that remain open in the field of Dataset Shift in Classification, shown in Section 7.

In Part II we provide a compendium of five publications that develop the goals presented, distributed in three sections:

- Unification of the study of Dataset Shift in Classification: Terminology and Experimental reviews.
- A proposal to solve Dataset Shift by means of Genetic Programming based Feature Extraction.
- Interactions between Dataset Shift and other classification issues: Imbalanced Datasets and k-fold Cross-Validation.

2. Preliminaries

This section includes a brief background relevant to the topics tackled in this thesis memory. In Subsection 2.1 we define some basic concepts about Dataset Shift, introduce k-fold cross-validation in Subsection 2.2, and finish with some notions about imbalanced classification in Subsection 2.3.

2.1. Dataset shift

Dataset shift is defined as the situation where the data used to train the classifier and the environment where said classifier is deployed do not follow the same distribution.

The first consideration when studying Dataset Shift is the characterizations of the different types of shifts that can appear and how these are generated. For the definitions used in this section, assume X represents the covariates of the problem and Y the class label. We also use the problem categorization proposed in [FF05], according to which there are two kinds of problems:

- X→Y problems, where the class label is causally determined by the values of the covariates. A typical example would be credit card fraud detection, since the behavior of the user, represented in the covariate space X, determines the class label: whether there is fraud or not.
- Y→X problems, where the class label causally determines the values of the covariates. Medical diagnosis usually falls in this category, where the disease, which is modeled as the class label Y, determines the symptoms, represented in the machine learning task as covariates X.

There are four different types of shift, depending on which probabilities change or not:

- **Covariate shift** appears only in X→Y problems, and is defined as the case where $P_{tr}(y|x) = P_{tst}(y|x)$ and $P_{tr}(x) \neq P_{tst}(x)$.
- **Prior probability shift** appears only in Y→X problems, and is defined as the case where $P_{tr}(x|y) = P_{tst}(x|y)$ and $P_{tr}(y) \neq P_{tst}(y)$.
- **Concept shift** is defined as
  - $P_{tr}(y|x) \neq P_{tst}(y|x)$ and $P_{tr}(x) = P_{tst}(x)$ in X→Y problems.
  - $P_{tr}(x|y) \neq P_{tst}(x|y)$ and $P_{tr}(y) = P_{tst}(y)$ in Y→X problems.

Some of the most common causes for the appearance of dataset shift are the different types of sample selection bias, and also domain shift. They are defined as:

- **Sample Selection Bias: Missing at Random (MAR).** MAR occurs when the probability of sampling an example (that is, including it in the training set) depends on x, but is independent on y.
- **Sample Selection Bias: Missing Not at Random (MNAR).** MNAR is the case where the sampling probability is both dependent on x and y.
- **Sample Selection Bias: Missing at Random - Class (MARC).** MARC appears when the sampling probability depends exclusively on y.

- **Domain Shift (DS).** This shift appears when there is a change in the scale of one or more of the attributes in x.

While sample selection bias and domain shift can be considered the most typical sources of dataset shift in real-world problems, a number of other causes exist, such as source component shift [ARJ08](a class is composed of several subclasses, and the prior probabilities of said subclasses change), adversarial environments [LL10](for instance, a hacker trying to break a security measure will try to disguise himself to be as different as possible from previous hackers), imbalanced classification problems [MTH10] (where rare examples or small disjuncts greatly increase the impact of even very soft degrees of dataset shift), or even dataset shift artificially introduced in a cross-validation setup [MTSH12].

2.2. k-fold cross-validation

Cross-validation [Koh95a] is a technique used for assessing how a classifier will perform when classifying new instances of the task at hand. One iteration of cross-validation involves partitioning a sample of data into two complementary subsets, training the classifier on one subset (called the training set), and testing its performance on the other subset (test set).

In k-fold cross-validation, the original sample is randomly partitioned into k subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the classifier, and the remaining $k-1$ subsamples are used as training data. The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as the test data. The k results from the folds are then averaged to produce a single performance estimation.

Cross-validation has been the subject of profuse study in the literature, some of the most interesting and relevant results are listed here:

- Repeated iterations of cross-validation asymptotically converge to a correct estimation of classifier performance [Sto77].
- 10-fold cross validation is better than leave-one out validation for model selection, and also better than other k-fold options [Koh95b].
- k-fold cross validation tends to underestimate classifier performance [Koh95b].

A basic, "dumb" k-fold cross-validation procedure would introduce Dataset Shift. This thesis studies this issue and how to avoid it in Section 5.3.2.

2.3. Imbalanced classification

A dataset is considered imbalanced when there is a large difference between the number of examples of each class. The study of classification in imbalanced domains is currently a very hot topic of research [HG09, SWK09], and there are studies showing its negative effect on classifier performance [XQ07]. There are two main approaches to combat the problem: cost-sensitive algorithms [Elk01, SKWW07], and data preprocessing [CBHK02, GH09].

Regarding the second option, it has been shown that applying a preprocessing step in order to balance the class distribution is a positive solution to the problem of imbalanced data-sets [BPM04, FGdJH08]. Furthermore, the main advantage of these techniques is that they are independent of the classifier used.

The problem of classifying data with high imbalance ratios remains open because the imbalance is not only a complication itself (in fact, the preprocessing methods presented in the literature would be capable of solving that perfectly); but also a factor that raises the importance of other issues. There are studies suggesting the complexity of an imbalanced classification problem is more closely correlated to the overlapping between classes than it is to the imbalance ratio [GMS08, DT10], and this work also explores the interaction between Dataset Shift and imbalance in Section 5.3.1.

3. Justification

To justify the relevance of this research work, we would like to address two different factors:

- **Relevance of the problem:** The issue of dataset shift appears often on real world data mining applications, mostly due to *sample selection biases* [CK05, Hec79, Zad04] when obtaining the training data. For example, a person who suspects he might be ill is more likely to get a medical test done; and it is wrong to assume that the ratio of a certain illness found in a hospital is representative of the presence of said illness among the general population.

  Other common causes of dataset shift include *adversarial* environments [BFR10, DDM+04, LL10] such as spam detection and fraud detection, *adversaries* continually adapt the test data to the output of the classification algorithm. The adversaries try to produce data (with some constraints) which the learner will misclassify as often as possible. This tends to produce general dataset shift as the adversary may alter the test distribution arbitrarily. In *non-stationary* environments, the dataset shift arises from a significant physical or temporal difference between training and test data sources. If a model trained on one continent is applied on another, for example, arbitrary changes in data distribution may result.

- **Degree of development of the research field:** The field of dataset shift is at a very attractive state due to its relatively young age (the earlier publications date back to 2000) and the raising interest it has received lately from several research groups. It remains a very open field, with plenty of opportunities for new developments, which are in turn useful for a growing community.

  The topic is also really interesting as a tool to enhance the understanding of other neighboring issues, an aspect of research we have begun with this work but that remains wide open and holds great promise.

4. Objectives

The aim of this thesis is to perform an in-depth study of Dataset Shift in Classification, establishing the basis for a more efficient and shareable research on the field, and also investigating how other complexity issues in Classification interact with it. To achieve these aims, we have set the following objectives:

- To establish a common terminology to define the concepts in the field, disambiguating the scattered literature. This common terminology is paramount to any serious efforts for the development of new theories and methods in the field.
- To create a base benchmark that would allow a fair comparison among different proposals. Since each author has, so far, used their own data to test their proposals, it is hard to know what works best under which conditions. A benchmark similar to what the UCI databases have accomplished is thus necessary.
- To compare the performances of state-of-the-art methods for Dataset Shift, shedding some light into which methods should be used in what circumstances.
- To propose an original method that is capable of solving general Dataset Shift problems.
- To study the interaction between Dataset Shift and k-fold cross-validation, proposing if necessary an alternative method do perform k-fold cross-validation that avoids the involuntary introduction of Dataset Shift into the experiments.
- To analyze the extent to which Dataset Shift and class imbalance ratios interact with each other, steeply increasing the difficulty of a given Classification problem.

5. Joint Discussion of Results

This section shows a summary of the different proposals presented in this dissertation, and it presents a brief discussion about the obtained results by each one.

5.1. Unification of the study of Dataset Shift in Classification: Terminology and Experimental reviews

In Section 2.1 we have introduced the basic concepts in dataset shift. The definitions presented in said section were not standard in the literature in 2010, with each author using their own definitions and terminology. For this reason, a terminology standardization was needed, and the conclusions reached when studying this issue can be found in subsection 5.1.1. Once the terminology is established, a comparison of the different proposals in the literature is closer. This topic, including the creation of a benchmark and the testing of the state-of-the-art solutions, is presented in subsection 5.1.2.

5.1.1. A unifying view on dataset shift in classification

Researchers studying the general problem of dataset shift, or specific instances of this problem, have coined a number of different names for it. These include *concept shift* [WK96], *concept drift* [WK96], *covariate shift* [Shi00], *data fracture* [CC09, MTLGB10] *reject inference* [Han98, CB04], and *imprecise class distributions* [ARGCCS07], among others. Worse still, researchers have sometimes used different terms to refer to the same problem, or given different definitions to the same term. To clear up this confusion and to make future research easier, we have carefully studied the terminology used in the literature and proposed a common convention which attempts to capture the essence of the terms as they are most commonly used. Specifically, we propose:

- *Covariate shift* if $P_{tst}(x) \neq P_{tr}(x)$ but $P_{tst}(y|x) = P_{tr}(y|x)$, in accordance with [Shi00].
- *Prior probability shift* if $P_{tst}(y) \neq P_{tr}(y)$ but $P_{tst}(y|x) = P_{tr}(y|x)$.
- *Concept shift* if $P_{tst}(x) = P_{tr}(x)$; but $P_{tst}(y|x) \neq P_{tr}(y|x)$ (in X→Y problems) or $P_{tst}(x|y) \neq P_{tr}(x|y)$ (in Y→X problems).
- *Dataset shift* if $P_{tst}(x, y) \neq P_{tr}(x, y)$ but none of the above hold.

Next, we surveyed common causes of dataset shift. *Sample selection bias* [CK05, Hec79, Zad04] occurs when the training sample is selected non-uniformly at random from the test population. Depending on the selection criteria and the type of classification problem, selection bias may produce covariate shift, prior probability shift, or general dataset shift. In *adversarial* environments [BFR10, DDM$^+$04, LL10] such as spam detection and fraud detection, *adversaries* continually adapt the test data to the output of the classification algorithm. The adversaries try to produce data (with some constraints) which the learner will misclassify as often as possible. This tends to produce general dataset shift as the adversary may alter the test distribution arbitrarily. In *non-stationary* environments, the dataset shift arises from a significant physical or temporal difference

between training and test data sources. If a model trained on one continent is applied on another, for example, arbitrary changes in data distribution may result.

5.1.2. Tackling Dataset Shift in Classification: Benchmark and Methods

Before a systematic experimental review can be carried out, the following issue needs to be addressed: the lack of a common set of benchmark problems that include different degrees and types of dataset shift. In this work, we propose a systematic methodology for the generation of artificially shifted datasets, which solves one of the main problems in the field and permits a fair comparison of solutions, which is paramount for the advancement of the field. Specifically, we have created used the following sources to introduce shift:

- MAR is implemented by having three different types of MAR: Top %, Gaussian and an Interval-based function, applied over some attribute in X, that would decide whether an example gets included in the final training set or not.
- MNAR is implemented in the following way: For all examples of the positive class, introduce a MAR bias; for all examples of the negative class, introduce a different MAR bias.
- MARC is implemented by assigning a different selection probability to examples of the positive class than those of the negative class.
- Domain shift is implemented as a linear rescaling of a single attribute in X.

The parameters used can be found in Table I.1, and 4 different degrees of shift (see Table I.2) for each source were created. For each dataset, 25 new datasets were created using each combination of source and degree of shift, resulting in $6 * 4 * 25 = 600$ shifted datasets for each original one.

Source of Dataset Shift	Parameter	Range	Description
Gaussian MAR	mean	$[min(x), max(x)]$	Mean of the normal dist.
	std	$[0{,}1 * (max(x) - min(x)),$ $0{,}2 * (max(x) - min(x))]$	Standard deviation of the normal distribution
Interval MAR	Interval	$[min(x), max(x)]$ [1]	Accept examples in the interval, reject the rest
TopN % MAR	N	$[0, 100]$	Accept the top N % examples, ranked according to attribute x
MNAR	biasPos	Any MAR Bias	Apply to negative class
	biasNeg	Any MAR Bias	Apply to positive class
MARC	p0	$[0, 1]$	$p(s\|y = 0)$
	p1	$[0, 1]$	$p(s\|y = 1)$
Domain Shift	mult	[0,1, 10]	$f(x) = x * mult + add$
	add	$[-mult * (max(x) - min(x))/2,$ $mult * (max(x) - min(x))/2]$	

Table I.1: Parameters needed by each dataset shift generator. x is the attribute along which the shift is injected.

The datasets created are available for download at http://sci2s.ugr.es/dataset-shift/.

Once the datasets were created, we were capable of testing the impact dataset shift has on the performance of some traditional classifiers, and we found that most classical classifiers can absorb a low degree of dataset shift without much loss in performance, but that once the degree of shift increases classifier performance is significantly affected.

Lastly, we tested several state of the art proposals against our benchmark datasets:

[1] Several intervals are generated, for example $[0{,}2, 0{,}3] \cup [0{,}56, 0{,}73]$

Degree of shift	Minimum % of examples selected	Maximum % of examples selected
Low Shift	70	98
Medium Shift	40	60
High Shift	25	37
Extreme Shift	10	20

Table I.2: Degrees of shift

- **Importance Weighted Cross Validation (IWCV)** [SKM07, KHS09]. Applies different weights to each sample to balance out data distribution in the presence of covariate shift.
- **Integrated Optimization Problem (IOP)** [BBS09]. Treats the learning under covariate shift as an integrated optimization problem, whose instantiation leads to a kernel logistic regression and an exponential model classifier for covariate shift.
- **Kernel Mean Matching (KMM)** [GSH$^+$09]. Reweighs training data to even the distribution with test data by matching covariate distributions in a high dimensional space.
- **SUbclass RE-estimation (SCRE)** [ARGCCS11]. This method, which does not require labels for the test set, was designed to tackle source component shift; but is also capable of dealing with other types. The idea behind it is to reestimate prior probabilities based on the different subclass distribution of the test set; which is obtained by the application of a clustering method.
- **GP-RFD** [MTLGB10]. Requires the test set to be partially labeled. Uses the performance of a classifier built on the training set over the labeled examples of the test to drive a Genetic-Programming based evolution that designs a transformation of the test set into a new one where the old classifier (the one that was built over the training set) has the best possible performance.

We found that SCRE performs clearly better than the other studied methods under almost all the conditions tested, and should be considered from now on the state of the art to measure new proposals against. The remaining studied methods perform quite poorly, which leads us to think there is still a large amount of improvement to be made in the field.

The journal articles associated to this part are:

- J.G. Moreno-Torres, T. Raeder, R. Aláiz-Rodríguez, N.V. Chawla, F. Herrera, A unifying view on dataset shift in classification. **Pattern Recognition, 45:1 (2012) 521-530. doi: 10.1016/j.patcog.2011.06.019.**
- J.G. Moreno-Torres, T. Raeder, R. Aláiz-Rodríguez, N.V. Chawla, F. Herrera, Tackling Dataset Shift in Classification: Benchmark and Methods. **Submitted to IEEE Transactions on Neural Networks and Learning Systems.**

5.2. A proposal to solve Dataset Shift by means of Genetic Programming based Feature Extraction (GP-RFD)

As one of the goals of this work, we focused on the development of an original solution to the general dataset shift problem.

We have proposed an algorithm that attempts to do it by means of a Genetic Programming-based method that performs feature extraction on the problem dataset driven by the accuracy of the previously built classifier; which we have named GP-RFD. The basic idea behind this method is shown in Figure 1, where Dataset A corresponds to the training data, Dataset B is the test data (partially labeled), and Dataset S is the modified test data, that can now be fed to the classifier trained over Dataset A.

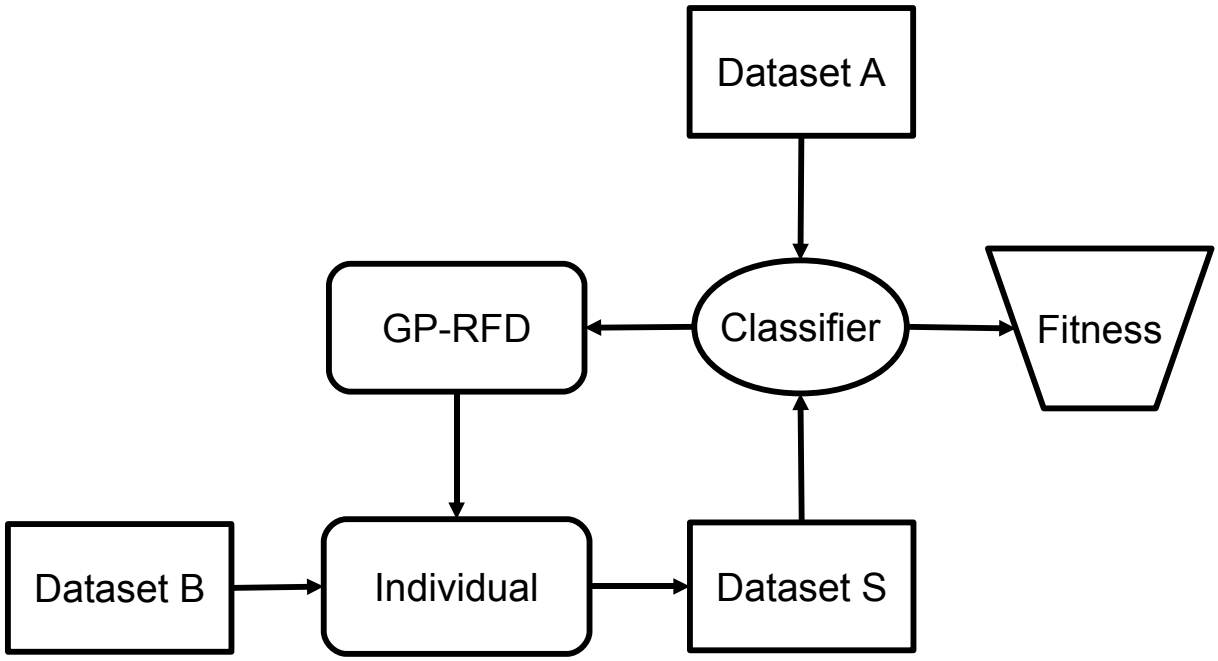


Figure 1: Schematic representation of the GP-RFD method

We have tested GP-RFD on a set of artificial benchmark problems, where a problem dataset is fabricated by applying an ad hoc disruption to an original dataset, and it has proved capable of solving all the transformations presented showing good performance both in train and, more importantly, test data.

We have also being able to apply GP-RFD to a real-world problem where data from two different laboratories regarding prostate cancer diagnosis was provided, and where the classifier learned from one did not perform well enough on the other. Our algorithm was capable of learning a transformation over the second dataset that made the classifier fit just as well as it did on the first one. The validation results with 5-fold cross validation also support the idea that the algorithm is obtaining good results; and has a strong generalization power.

Lastly, we have applied a statistical analysis methodology that supports the claim that the classifier performance obtained on the solution dataset significantly outperforms the one obtained on the problem dataset.

The journal article associated to this part is:

- J.G. Moreno-Torres, X. Llorà, D.E. Goldberg, R. Bhargava, Repairing fractures between

ta using Genetic Programming-based feature extraction: A case study in cancer diagnosis. **Information Sciences, 222 (2013) 805-823. doi: 10.1016/j.ins.2010.09.018.**

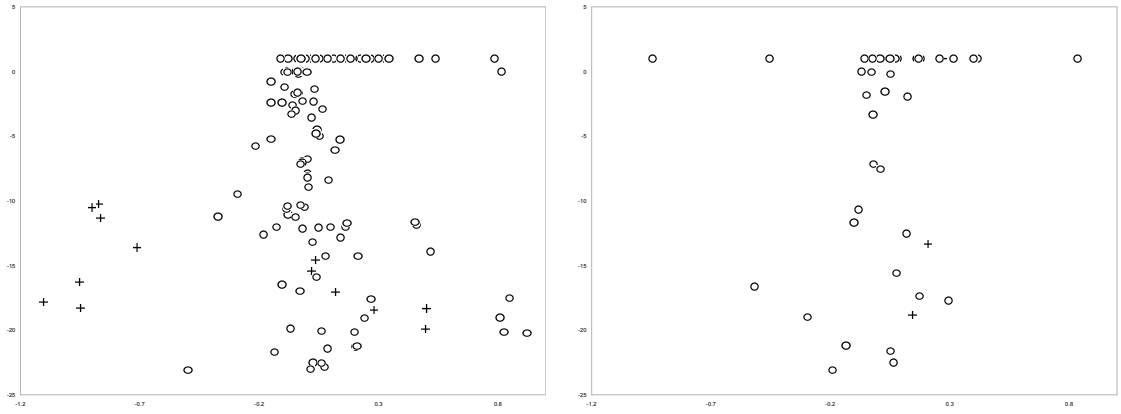
5.3. Interactions between Dataset Shift and other classification issues: Imbalanced datasets and k-fold cross-validation

This section details the conclusions learned when studying the interactions between Dataset Shift and other classification issues. Imbalanced datasets are covered in subsection 5.3.1; while k-fold cross-validation is presented in subsection 5.3.2.

5.3.1. Interaction between Dataset Shift and imbalanced datasets

We have presented GP-RST, a GP-based feature extractor that employs RST techniques to estimate the fitness of individuals. We have shown GP-RST to be a competitive preprocessing method for highly imbalanced datasets, with the added advantage of providing bidimensional representations of the datasets it preprocesses, which are easily interpreted.

We have, through the analysis of the visual representations of the preprocessed datasets, observed a dataset shift incidence between training and test sets, specially in the minority class, that is affecting the classification performance. An example of this behavior is shown on Figure 2.



(a) Training set (0.6794) (b) Test set (0.0000)

Figure 2: Example of bad classifier performance due to dataset shift, dataset glass_2. Classifier performance in parenthesis.

We believe this discovery is very relevant since it challenges the usual assumptions when experimenting with preprocessing for highly imbalanced data.

The conference contribution associated to this part is:

- J.G. Moreno-Torres, F. Herrera, A Preliminary Study on Overlapping and Data Fracture in Imbalanced Domains by means of Genetic Programming-based Feature Extraction. **Proceedings of 10th International Conference on Intelligent Design and Applications (ISDA), 2010, pages 501-506.**

5.3.2. Interaction between Dataset Shift and k-fold cross-validation

We have presented an experimental analysis on the impact covariate shift introduced by partitioning can have on the reliability of classifier performance evaluation through cross-validation and shown that, when covariate shift is introduced, single-experiment reliability is diminished and the number of iterations required to reach a stable state is increased.

We have found that cross-validation approaches that try and limit the impact of partition-induced covariate shift are more reliable when running a single experiment, and need a lower number of iterations to stabilize. Among them, we have shown that DOB-SCV is more effective than DB-SCV, and thus recommend cross-validation users to use DOB-SCV as the partitioning method in order to avoid covariate-shift related problems.

We have studied the number of iterations needed to reach a stable performance estimation with the different partitioning strategies, finding that DOB-SCV outperforms the others, also supporting the recommendation of DOB-SCV as the best partitioning method to avoid covariate-shift related issues.

The journal article associated to this part is:

- J.G. Moreno-Torres, J.A. Sáez, F. Herrera, Study on the impact of partition-induced dataset shift on k-fold cross-validation. **IEEE Transactions on Neural Networks and Learning Systems, 23:8 (2012) 1304-1312. doi: 10.1109/TNNLS.2012.2199516.**

6. Conclusiones

Mi primer contacto con el campo de la Fractura de Datos se dió al intentar resolver un problema real: un clasificador para el diagnóstico de cáncer de próstata construido con los datos de un laboratorio sufría un gran descenso de precisión al ser aplicado para clasificar datos de otro laboratorio. Una búsqueda preliminar en la literatura relacionada mostró un campo que estaba en un estado temprano de desarrollo, en el que los esfuerzos hechos por unos pocos investigadores estaban descoordinados e incluso la terminología básica sufría de una clara falta de estandarización. Por lo tanto, hicimos lo que el resto de los investigadores habían hecho hasta ese momento: diseñar una solución a medida para el problema que teníamos entre manos, y probarla en unos pocos problemas artificiales; obteniendo resultados positivos pero sin llegar nunca a compararla contra otros métodos diseñados para problemas parecidos.

Parte de la comunidad comenzó entonces un intento de resolver estas cuestiones [QnCSSL09], pero muchos detalles quedaban abiertos, por lo que este trabajo se enfocó a establecer un marco común para futuros desarrollos de investigación en el campo.

El primer paso en esta dirección es la estandarización de la terminología, una tarea que hemos concluido con éxito al definir claramente cada tipo de Fractura de Datos, así como varias de las diversas causas potenciales.

Segundo, la propuesta de una serie de conjuntos dedatos con distintos tipos y grados de fractura es una herramienta que futuros proyectos de investigación en la materia encontrarán sin duda útil. El hecho de haber llevado a cabo una comparación exhaustiva de las propuestas del estado del arte utilizando estos conjuntos de referencia los convierten en un recurso completo para la comparación de nuevas propuestas.

Una vez tuvimos una idea clara de cómo la fractura de datos afecta a la clasificación, estábamos en una buena posición para desafiar algunas de las suposiciones establecidas acerca del procedimiento de clasificación. Esto llevó a un estudio de la fractura covariada introducida al realizar validación cruzada, donde se llegó a una conclusión clara: las técnicas de particionamiento que no tienen en cuenta la fractura covariada son menos fiables en cuanto a la estimación del comportamiento de los clasificadores.

Por último, un estudio sobre la relación entre fractura de datos y clasificación no balanceada confirmó una sospecha que otros autores ya habían expresado: la clasificación no balanceada no es difícil por sí misma, pero actúa como catalizador incrementando el efecto de algunos factores de complejidad como solapamiento o ruido. Hemos mostrado que la fractura de datos pertenece a esa lista de factores de complejidad, y que debe ser tenida en cuenta en problemas de clasificación genéricos, pero más todavía cuando existe un alto desequilibrio entre las clases.

6. Concluding remarks

My first contact with the field of Dataset Shift came from trying to solve a real-world problem: a classifier to diagnose prostate cancer built over data from one laboratory was suffering a huge drop in performance when used to classify data from a different laboratory. A preliminary literature research showed a field that was in a very preliminary stage of its development, with the few efforts being done by different researchers being uncoordinated and even the basic terminology having a glaring lack of standardization. Therefore, we did what all the other researchers had done up to that point: design an *ad-hoc* solution for the specific problem at hand, and test it over a few artificial problems, obtaining positive results but never testing it against methods designed for similar problems.

An attempt to solve these issues was begun by part of the community [QnCSSL09], but a lot of questions remained open, and that is why this work focused on establishing a common framework for future research developments on the field.

The first step in this direction is the standardization of the terminology, a task we have successfully accomplished by clearly defining each type of shift and several different potential causes.

Secondly, the proposal of a common benchmark of datasets with different types and degrees of dataset shift is a basic tool that future research projects on the topic will undoubtedly find useful. The fact that we have also carried out an exhaustive comparison of the state-of-the-art proposals using this benchmark makes it a complete resource for the testing of new proposals.

Once we had a clear picture of how dataset shift affects classification, we were in a good position to challenge some established assumptions about the classification procedure. This led to a study on partition-based covariate shift, where a clear conclusion was found: partition techniques that do not take covariate shift into account are less reliable in terms of classifier performance estimation.

Lastly, an investigation on the relationship between dataset shift and imbalanced classification confirmed a suspicion other authors had already hinted at: imbalance classification is not difficult *per se*, but it acts as a catalyst enhancing the effect of some complexity factors such as overlap or noisy data. We have shown that dataset shift belongs in that list of complexity factors, and that it needs to be taken into account in general classification tasks, but more so in those where there is a high imbalance between the classes.

7. Future Work

As we mentioned in Section 3, the field of Dataset Shift is still in the early stages of its development. While we believe this work represents an important step forward, there are still several interesting avenues to pursue. Amongst them, the most promising ones are:

- Study and comparison of the **dataset shift detection** methods proposed in the literature: The experimental review included in this work focuses on dataset shift solvers, but there are a handful of proposals for the detection and characterization of dataset shift in the literature (see [WZF+03, YWZ08, CC09]); and it would be interesting to test them and, if appropriate, propose an alternative method.
- Dataset shift in classification has a lot of characteristics in common with other deeply studied issues. The relationship between said fields and dataset shift could be a source of inspiration for new developments in the field. Specifically, it would be interesting to:

  1. **Apply noise cleaning techniques** to data that suffers from dataset shift, with the idea that eliminating examples that "do not make sense" in the training data, the model built would be more general and thus robust in the face of dataset shift.
  2. **Explore transfer learning methods** and their application to dataset shift problems. Even though the premises are not the same, the underlying problem transfer learning and dataset shift tackle is: data where model was built and where it is deployed are, for whatever reason, different. Applying state-of-the-art transfer learning algorithms to dataset shift problems seems like the next logical step under this understanding.
  3. **Apply ensemble classifier solutions** to dataset shift classification problems. Ensemble classifiers [Die00] have proven to be a highly robust approach to complex classification problems. An ensemble of classifiers trained with subsets of the training data has the potential to maintain a good performance even when faced with high levels of dataset shift.
  4. **Solve Big Data problems** by applying dataset shift detection techniques to enhance instance selection procedures, so that the subsampled data is as representative as possible of the general Big Data problem.

Part II. Publications: Published, Accepted and Submitted Papers

1. Unification of the study of Dataset Shift in Classification: Terminology and Experimental reviews

The journal papers associated to this part are:

1.1. A unifying view on dataset shift in classification

- J.G. Moreno-Torres, T. Raeder, R. Aláiz-Rodríguez, N.V. Chawla, F. Herrera, A unifying view on dataset shift in classification. Pattern Recognition, 45:1 (2012) 521-530. doi: 10.1016/j.patcog.2011.06.019.
  - Status: **Published**.
  - Impact Factor (JCR 2011): 2.292.
  - Subject Category: Computer Science, Artificial Intelligence. Ranking 18 / 111 (Q1).
  - Subject Category: Engineering, Electrical & Electronic. Ranking 35 / 245 (Q1).

Pattern Recognition 45 (2012) 521–530

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A unifying view on dataset shift in classification

Jose G. Moreno-Torres [a,*], Troy Raeder [b], Rocío Alaiz-Rodríguez [c], Nitesh V. Chawla [b], Francisco Herrera [a]

[a] Department of Computer Science and Artificial Intelligence, Universidad de Granada, 18071 Granada, Spain
[b] Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA
[c] Universidad de León, Dpto. de Ingeniería Eléctrica y de Sistemas, Campus de Vegazana, 24071 León, Spain

ARTICLE INFO

Article history:
Received 29 November 2010
Received in revised form
6 June 2011
Accepted 15 June 2011
Available online 18 July 2011

Keywords:
Dataset shift
Data fracture
Changing environments
Differing training and test populations
Covariate shift
Sample selection bias
Non-stationary distributions

ABSTRACT

The field of dataset shift has received a growing amount of interest in the last few years. The fact that most real-world applications have to cope with some form of shift makes its study highly relevant. The literature on the topic is mostly scattered, and different authors use different names to refer to the same concepts, or use the same name for different concepts. With this work, we attempt to present a unifying framework through the review and comparison of some of the most important works in the literature.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

The machine learning community has analyzed data quality in classification problems from different perspectives, including data complexity [29,7], missing values [19,21,39], noise [11,64,58,38], imbalance [52,27,53] and, as is the case with this paper, dataset shift [4,44,14]. Dataset shift occurs when the testing (unseen) data experience a phenomenon that leads to a change in the distribution of a single feature, a combination of features, or the class boundaries. As a result the common assumption that the training and testing data follow the same distributions is often violated in real-world applications and scenarios.

While the research area of dataset shift has received significant attention in recent years (most of the work is published in the last eight years), the field suffers from a lack of standard terminology. Independent authors working under different conditions use different terms, making it difficult to find and compare proposals and studies in the field.

Contributions. The main goal of this work is to provide a unifying framework through the review and analysis of some of the most important publications in the field, comparing the terminology used in each of them and the exact definitions that were given. We present a framework that can be useful in future research and, at the same time, provide researchers unfamiliar with the topic a brief introduction to it. Our goal with this work is to not only unify different methods and terminologies under a taxonomical structure, but also provide a guide to a researcher as well as a practitioner in machine learning and pattern recognition. We use the notation in [44] as the base for the comparisons. We also present a brief summary of solutions proposed in the literature.

The remainder of this paper is organized as follows: Some basic notation is introduced in Section 2. In Section 3, an analysis of the name given to the field of study is presented. Section 4 details the terminology used for the different types of dataset shift that can appear. Section 5 presents examples demonstrating the effect of these shifts on classifier performance. An analysis of some common causes of dataset shift is presented in Section 6. A brief summary of the solutions proposed in the literature is shown in Section 7. Finally, some conclusions are presented in Section 8.

2. Notation

In this work, we focus on the analysis of dataset shift in classification problems. A classification problem is defined by:

- A set of features or *covariates* x.
- A target variable y (the class variable).
- A joint distribution $P(y,x)$.

* Corresponding author.
E-mail addresses: jose.garcia.mt@decsai.ugr.es (J.G. Moreno-Torres), traeder@cse.nd.edu (T. Raeder), rocio.alaiz@unileon.es (R. Alaiz-Rodríguez), nchawla@cse.nd.edu (N.V. Chawla), herrera@decsai.ugr.es (F. Herrera).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.06.019

When analyzing dataset shift, the relationships between the covariates and the class label are particularly relevant. Fawcett and Flach [20] proposed a taxonomy to classify problems according to an intrinsic property of the data generation process: the causal relationship between class label and covariates. This particular characteristic of a problem determines what kinds of shift can affect a given problem, so the rest of the paper is structured regarding the two different kinds of problems generated by this distinction:

- $X \rightarrow Y$ problems, where the class label is causally determined by the values of the covariates. A typical example would be credit card fraud detection, since the behavior of the user, represented in the covariate space X, determines the class label: whether there is fraud or not.
- $Y \rightarrow X$ problems, where the class label causally determines the values of the covariates. Medical diagnosis usually falls in this category, where the disease, which is modeled as the class label Y, determines the symptoms, represented in the machine learning task as covariates X.

The joint distribution $P(y,x)$ can be written as

- $P(y|x)P(x)$ in $X \rightarrow Y$ problems.
- $P(x|y)P(y)$ in $Y \rightarrow X$ problems.

In this prototypical classification problem, the output of the system or learning algorithm takes on N (symbolic) values $y = \{1, \ldots, N\}$ corresponding to N classes. A commonly used loss function for this problem measures the classification error

$$L(y, f(x,\omega)) = \begin{cases} 0 & \text{if } y = f(x,\omega) \\ 1 & \text{if } y \neq f(x,\omega) \end{cases}$$

where ω denotes the set of classifier parameters. Using this loss function, the risk functional

$$R(\omega) = \int L(y, f(x,\omega)) p(x,y)\, dx\, dy$$

quantifies the probability of misclassification. Learning then becomes the problem of estimating the function $f(x,\omega_0)$ (classifier) that minimizes the probability of misclassification using only the training data.

When we use the terms *training* and *test* stages, we refer to the data available to train the classifier and the data present in the environment the classifier will be deployed in, respectively. The data distributions in training and test are denoted as P_{tr} and P_{tst}.

3. Dataset shift

The term "dataset shift" was first used in the book by Quiñonero-Candela et al. [44], the first compilation on the field, where it was defined as "cases where the joint distribution of inputs and outputs differs between training and test stage" [49].

One of the main problems in the field is the lack of visibility most works suffer, since there is not even a standard term to refer to it. So far, each author has chosen a different name to refer to the same basic idea. As an example, the following terms have been used in the literature to refer to dataset shift:

- "Concept shift" or "concept drift" [57,17], where the idea of different data distributions is associated with changes in the class definitions (i.e. the "concept" to be learned).
- "Changes of classification" [55], where it is defined as "In the change mining problem, we have an old classifier, representing some previous knowledge about classification, and a new data set that has a changed class distribution."
- "Changing environments" [4], defined as "The fundamental assumption of supervised learning is that the joint probability distribution $p(x\|d)$ will remain unchanged between training and testing. There are, however, some mismatches that are likely to appear in practice."
- "Contrast mining in classification learning" [60], a slightly different take on the issue: "Given two groups of interest, a user often needs to know the following. Do they represent different concepts? To what degree do they differ? What is the discrepancy and where does it originate from?"
- "Fracture points", defined in [14] as "fracture points in predictive distributions and alteration to the feature space, where a fracture is considered as the points of failure in classifiers' predictions - deviations from the expected or the norm."
- "Fractures between data", used in [40], defined as the case where "we have data from one laboratory (dataset A), and derive a classifier from it that can predict its category accurately. We are then presented with data from a second laboratory (dataset B). This second dataset is not accurately predicted by the classifier we had previously built due to a fracture between the data of both laboratories."

Such inconsistent terminology is a disservice to the field as it makes literature searches difficult and confounds the discussion of this important problem. We recommend the term *dataset shift* for any situation in which training and test data follow distributions that are in some way different. Formally, we define it as

Definition 1. *Dataset shift* appears when training and test joint distributions are different. That is, when $P_{tr}(y,x) \neq P_{tst}(y,x)$.

4. Types of dataset shift

In this section, we present an analysis of the different kinds of shift that can appear in a classification problem. Section 4.1 deals with covariate shift, while Sections 4.2 and 4.3 explain prior probability shift and concept shift, respectively. A graphical example is introduced to illustrate each of these cases. The section is closed with Section 4.4, where other potential types of shifts are explained.

4.1. Covariate shift

The term covariate shift was first defined ten years ago in [47] where it refers to changes in the distribution of the input variables x. Covariate shift is probably the most studied type of shift, but there appears to be some confusion in the literature about the exact definition of the term. There are also some equivalent names, such as "population drift" [31,26]. Some definitions of covariate shift found in the literature are:

- "Case when the population distribution can change over time" (this concept is defined as "population drift" in [31]).
- "Let x be the explanatory variable or the covariate, (…). Let $q_1(x)$ be the density of x for evaluation of the predictive performance, while $q_0(x)$ be the density of x in the observed data. The situation $q_0(x) \neq q_1(x)$ will be called covariate shift in distribution." [47].
- "Change in the data distributions" [26], uses the term 'population drift'.
- "The input distribution $p(x)$ varies but the functional relation $p(y|x)$ remains unchanged" [59].
- "Differing training and test distributions" [8], who define it as follows (the two definitions appear in different places in the same paper):
  - "The training instances are governed by a distribution that is allowed to differ arbitrarily from the test distribution."

  - Training and test distribution may differ arbitrarily, but there is only one unknown target conditional class distribution $p(y|x)$."
- "The conditional probability $p(y|x)$ remains unchanged, but the input distribution $p(x)$ differs from training to future data" [4].
- "The data distribution generating the feature vector x and its related class label y changes as a result of a latent variable t. Thus, we may state that covariate shift has occurred when $P(y|x,t1) \neq P(y|x,t2)$" [14].

The concept of covariate shift is not standardized enough, as can be seen from the differences between the definitions shown above. The definition given by Cieslak and Chawla [14] states that $P(y|x,t1) \neq P(y|x,t2)$, while Yamakazi et al. [59] or Alaiz-Rodríguez et al. [4] state that $p(y|x)$ remains unchanged. Even within the same paper, the two definitions given by Bickel et al. [8] are not equivalent.

In [49], covariate shift is defined as something that occurs "when the data is generated according to a model $P(y|x)P(x)$ and where the distribution $P(x)$ changes between training and test scenarios." This seems to capture the essence of the term as it is most commonly used. Thus, we propose the following as a consistent formal definition.

Definition 2. *Covariate shift* appears only in $X \rightarrow Y$ problems, and is defined as the case where $P_{tr}(y|x) = P_{tst}(y|x)$ and $P_{tr}(x) \neq P_{tst}(x)$.

The analogous issue in $Y \rightarrow X$ problems is prior probability shift, studied in Section 4.2.

Assume we have an $X \rightarrow Y$ problem where there is one covariate x_0 and a target y. The training data distribution $P_{tr}(x_0)$ is composed by the union of two Gaussian distributions with variance 0.5 (one with mean $x_0 = -2$ and the other with mean $x_0 = 2$) and $P_{tr}(y|x_0)$ is defined as

$$P_{tr}(y|x_0) = \frac{1}{1+\exp\left(\frac{-x_0}{0.2}\right)}$$

Consider now that in the test data, $P_{tst}(y|x_0)$ remains unchanged, but the Gaussian distributions that compose $P_{tst}(x_0)$ are now centered in $x_0 = -1$ and $x_0 = 1$, respectively. Fig. 1 depicts this simple example of covariate shift where $P_{tr}(x_0) \neq P_{tst}(x_0)$.

4.2. Prior probability shift

Prior probability shift refers to changes in the distribution of the class variable y. It also appears with different names in the literature, and the definitions have slight differences between them:

- "Change in class distributions" [56], the authors call it "varying class distributions".
- "The class prior probability $p(y)$ varies from training to test, but $p(x|y)$ remains unaltered" [4], denoted as "change in class distribution".
- "Shifting priors occurs when sampling is dependent on the class label and independent of the feature vector x" [14].

Storkey [49] defines prior probability shift as a case where "an assumption is made that a causal model of the form $P(x|y)P(y)$ is valid, (…), the distribution $P(y)$ changes between training and test situations." According to the definitions present in the literature, prior probability shift is the reverse case of covariate shift. More formally, we define it as

Definition 3. *Prior probability shift* appears only in $Y \rightarrow X$ problems, and is defined as the case where $P_{tr}(x|y) = P_{tst}(x|y)$ and $P_{tr}(y) \neq P_{tst}(y)$.

As an example, assume we have a $Y \rightarrow X$ problem with one covariate x_0 and a target y that may take the class values $y=0$ and $y=1$. In the training data, $P_{tr}(y=0) = P_{tr}(y=1) = 0.5$ and $P_{tr}(x_0|y)$ is defined as

$$x_0 = \begin{cases} \mathcal{N}(2,0.5) & \text{when } y = 1 \\ \mathcal{N}(-2,0.5) & \text{otherwise} \end{cases}$$

Consider now that in the test data, $P_{tst}(x_0|y=0)$ and $P_{tst}(x_0|y=1)$ remain unchanged, but the class prior probabilities vary, taking the values $P_{tst}(y=1) = 0.70$ and $P_{tst}(y=0) = 0.30$. This example is illustrated in Fig. 2.

Lastly, it is important to mention that prior probabilities are closely related to cost-sensitive learning [54], so techniques from that field are also applicable.

4.3. Concept shift

Concept shift is usually referred to as "concept drift" in the literature; we propose a change in name here for consistency with the above. Even though this type of shift was not mentioned in [44], some other authors have studied it and proposed the following definitions:

- "A changing context can induce changes in the target concepts, producing what is known as concept drift" [57].
- "A user's behaviors and tasks change with time" [34].
- "Changes to the definitions of the classes" [26].
- "$p(y|x)$ changes between the training and test phases" [59], the author used the term "functional relation change"
- "Case where $p(x)$ is not altered but, $p(y|x)$ varies from training to test" [4], denoted as "class definition change".

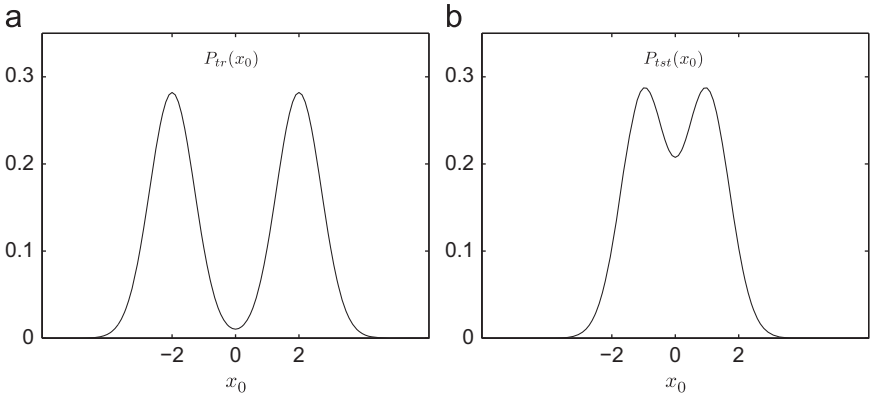


Fig. 1. Covariate shift: $P_{tst}(y|x_0) = P_{tr}(y|x_0)$ and $P_{tr}(x_0) \neq P_{tst}(x_0)$. (a) Training data and (b) test data.

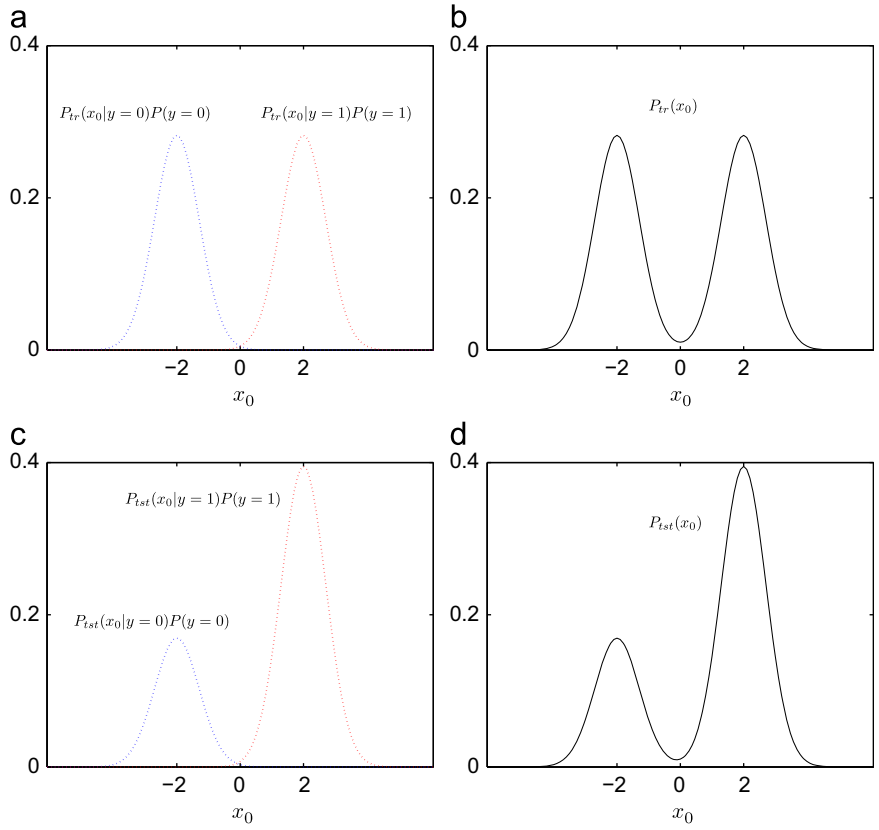


Fig. 2. Prior probability shift. Training dataset with $P_{tr}(y=0)=P_{tr}(y=1)=0.5$. Test dataset with $P_{tr}(y=0)=0.3$ and $P_{tr}(y=1)=0.7$. Class conditional data densities remain constant: $P_{tst}(x_0|y=0)=P_{tr}(x_0|y=0)$ and $P_{tst}(x_0|y=1)=P_{tr}(x_0|y=1)$. (a) Training data, (b) training data density, (c) test data and (d) test data density.

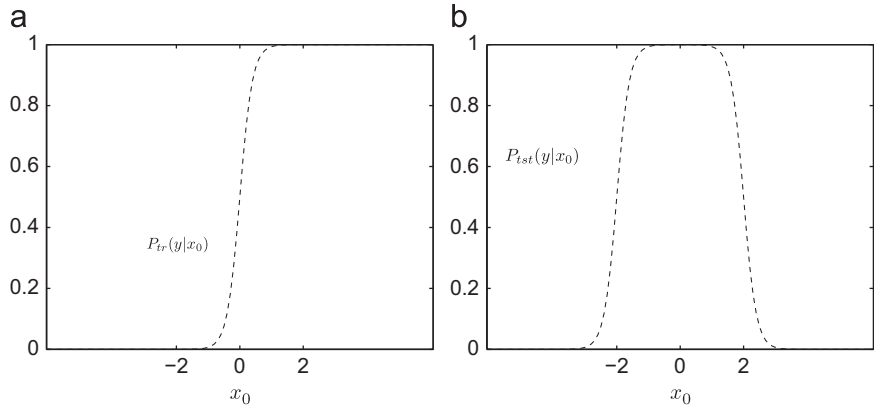


Fig. 3. Example of concept shift: data density remains constant $P_{tr}(x_0)=P_{tst}(x_0)$ and $P_{tr}(y|x_0)\neq P_{tst}(y|x_0)$. (a) Training set and (b) test set.

Concept shift happens when the relationship between the input and class variables changes, which presents the hardest challenge among the different types of dataset shift that has been tackled so far. Formally, we define it as

Definition 4. *Concept shift* is defined as

- $P_{tr}(y|x)\neq P_{tst}(y|x)$ and $P_{tr}(x)=P_{tst}(x)$ in $X\rightarrow Y$ problems.
- $P_{tr}(x|y)\neq P_{tst}(x|y)$ and $P_{tr}(y)=P_{tst}(y)$ in $Y\rightarrow X$ problems.

As an example of concept shift, consider the training dataset with the distribution presented for the covariate shift problem. If a concept shift takes place, the test set data distribution $P_{tst}(x_0)$ remains constant, but $P_{tst}(y|x_0)$ is redefined, for instance, as

$$P_{tst}(y|x_0)=\frac{1}{\left(1+\exp\left(\frac{-2+x_0}{0.2}\right)\right)\left(1+\exp\left(\frac{-2-x_0}{0.2}\right)\right)}$$

Fig. 3 shows the $P_{tr}(y|x_0)$ and $P_{tst}(y|x_0)$ for a concept shift problem.

4.4. Other types of dataset shift

Even though the shifts presented above are the most commonly present in real-world classification tasks, there are others

that could in theory also happen, included here for completeness:

- $P_{tr}(y|x) \neq P_{tst}(y|x)$ and $P_{tr}(x) \neq P_{tst}(x)$ in $X \rightarrow Y$ problems.
- $P_{tr}(x|y) \neq P_{tst}(x|y)$ and $P_{tr}(y) \neq P_{tst}(y)$ in $Y \rightarrow X$ problems.

There are two main reasons these shifts are usually not considered in the literature: they appear more rarely than the others and, most importantly, they are so hard that we currently consider them impossible to solve.

5. Examples of the relevance of dataset shift

The examples presented in Sections 4.1 and 4.2 were designed to showcase as clearly as possible what covariate and prior probability shift mean. However, they do not show why its study is important: the negative effect dataset shift often has on classifier performance.

This section presents new examples for both covariate shift and prior probability shift, where the said shifts actually produce a change in the Bayes error boundary.

Fig. 4 depicts a case of covariate shift where the shift produces a change in the Bayes error boundary resulting in a drop in the classifier performance. In this example, assume we have an $X \rightarrow Y$ problem where there is one covariate x_0 and a target class label y that takes the values $y=0$ and $y=1$. In the training data, $P_{tr}(x_0)$ is composed by the union of two Gaussian distributions, $\mathcal{N}(-1.5,0.5)$ and $\mathcal{N}(1.5,0.5)$, that are the data distributions of each class, respectively. In the test data, $P_{tst}(y|x_0)$ remains unchanged, but the Gaussian distributions that compose $P_{tst}(x_0)$ now have means -1.5 and 0.5, respectively. Fig. 4(d) shows the difference between the optimal decision boundary (continuous line) in the test set and that one estimated from the training dataset (dashed line).

Fig. 5, on the other hand, shows a case of prior probability shift. For this example, assume we have a $Y \rightarrow X$ problem with a covariate x_0 and a target y. In the training data, $P_{tr}(y=0) = P_{tr}(y=1) = 0.5$ and $P_{tr}(x_0|y)$ is defined as

$$x_0 = \begin{cases} \mathcal{N}(1.5,0.5) & \text{when } y=1 \\ \mathcal{N}(-1.5,0.5) & \text{otherwise} \end{cases}$$

In the test data, $P_{tst}(x_0|y)$ remains unchanged, but the prior probabilities change to $P_{tst}(y=1)=0.8$ and $P_{tst}(y=0)=0.2$. Fig. 5 illustrates this problem and Fig. 5(d) highlights the difference between the optimal decision boundary (continuous line) and the boundary estimated in the training stage. If the class prior probabilities differ from the ones assumed during learning, the classifier performance will be suboptimal.

6. Causes of dataset shift

In this section we comment on some of the most common causes of dataset shift. These concepts have created confusion at times, so it is important to remark that these terms are factors that can lead to the appearance of some of the shifts explained in Section 4, but they do not constitute dataset shift themselves.

There are several possible causes for dataset shift, out of which this section mentions the two we deem most important: Sample selection bias and non-stationary environments. In the first one, the discrepancy in distribution is due to the fact that the training examples have been obtained through a biased method, and thus do not represent reliably the operating environment where the classifier is to be deployed (which, in machine learning terms, would constitute the test set). This case is studied in Section 6.1, and is the one most commonly analyzed in the literature.

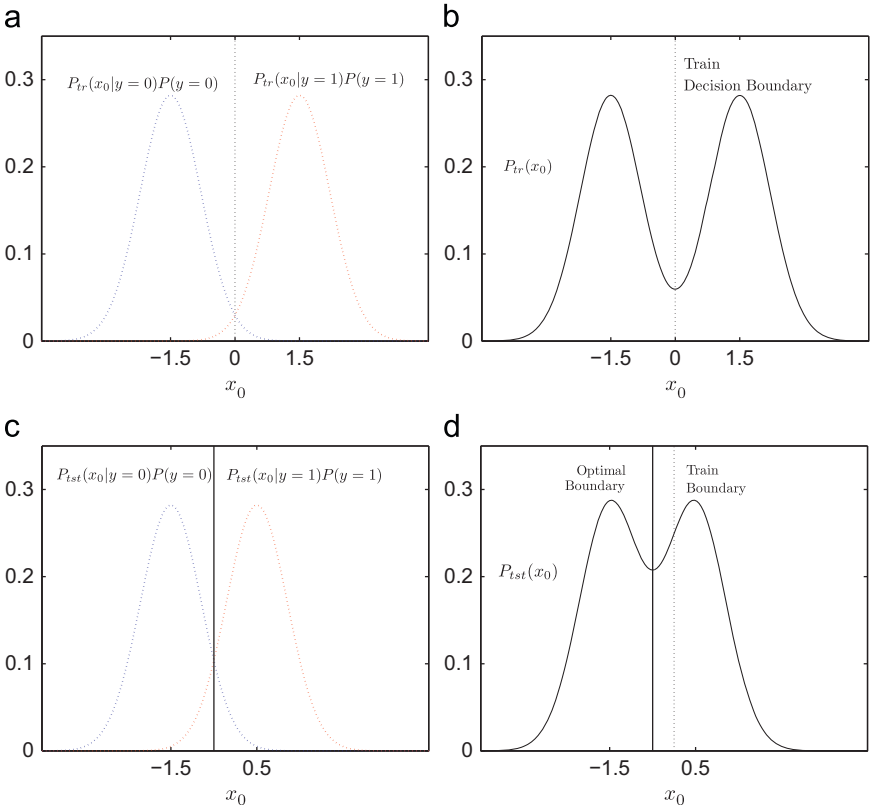


Fig. 4. Example of covariate shift with an influence on the Bayes error boundary. The vertical dotted line represents the boundary learned by the classifier using the training set. The vertical continuous line represents the optimal boundary for the test set. (a) Training set, (b) training data density, (c) test set and (d) test data density.

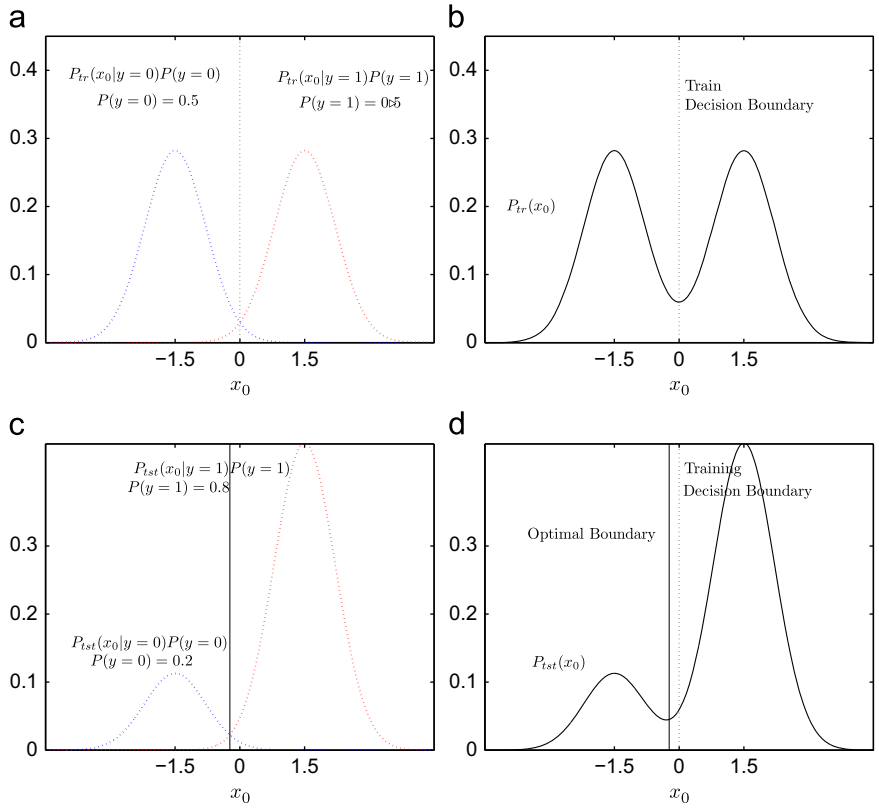


Fig. 5. Example of prior probability shift with an influence on the Bayes error boundary. The vertical dotted line represents the boundary learned by the classifier using the training set. The vertical continuous line represents the optimal boundary for the test set. (a) Training set, (b) training data density, (c) test set and (d) test data density.

A typical example of this case would be the analysis of a process where, due to cost concerns, one of the classes is sampled at a lower rate than it actually appears.

The second cause appears when the training environment is different from the test one, whether it is due to a temporal or a spatial change. It commonly appears, among others, in adversarial classification problems; and it is analyzed in Section 6.3.

6.1. Sample selection bias

The term *sample selection bias* refers to a systematic flaw in the process of data collection or labeling which causes training examples to be selected *non-uniformly* from the population to be modeled. In social science research, for example, there will be subsets of the general population (students at the researcher's University or previous study participants) which are easier to survey than others. These "easy" populations may be over-represented in the training sample, whereas "difficult" populations (i.e. prisoners) may be under-represented or completely excluded.

One can imagine any number of permutations of this general problem. If data are collected from remote sensors, for example, the different sensors may malfunction at different rates or collect data at different rates, meaning that certain portions of the observation area are over-represented.

The problem of operating under sample selection bias has received substantially more attention in other domains than it has in the machine learning community. In the credit scoring literature it goes by the name of *reject inference*, because potential credit applicants who are *rejected* under the previous model are not available to train future models [15,25].

The term has been used as a synonym of covariate shift [30] (which is not correct, as was stated above), but also on its own as a related problem to dataset shift. In that line, Storkey [49] proposes the following formal definition:

Definition 5. *Sample selection bias*, in general, causes the data in the training set to follow $P_{tr}=P(s=1|x,y)$, while the data in the test set follows $P_{tst}=P(y,x)$. Depending on the type of problem, we have:

- $P_{tr}=P(s=1|y,x)P(y|x)P(x)$ and $P_{tst}=P(y|x)P(x)$ in $X\rightarrow Y$ problems,
- $P_{tr}=P(s=1|y,x)P(x|y)P(y)$ and $P_{tst}=P(x|y)P(y)$ in $Y\rightarrow X$ problems,

where s is a binary selection variable that decides whether a datum is included in the training sample process ($s=1$) or rejected from it ($s=0$).

In [37,61,14], three different types of sample selection bias were analyzed:

Definition 6. *Missing completely at random (MCAR)* occurs when the sampling method is completely independent of x and y, so that $P(s=1|y,x)=P(s=1)$. This kind of bias does not produce any dataset shift.

Definition 7. *Missing at random (MAR)* occurs when s depends on x but conditional on x is independent of y; so that $P(s=1|y,x)=P(s=1|x)$. This kind of bias can potentially produce covariate shift.

To illustrate more clearly the relationship between MAR bias and covariate shift, note that one can "correct" for covariate shift when estimating model performance by using *importance-weighted cross-validation* [51]. That is to say, an *unbiased estimate* of the classification loss on a set of feature vectors x_i and their associated classes y_i can be obtained by weighting the loss associated with each x_i by $P_{tst}(x_i)/P_{tr}(x_i)$. More formally, if the k-fold cross-validation

estimate of misclassification cost is given by

$$\frac{1}{k}\sum_{j=1}^{k}\frac{1}{|\mathcal{F}_j|}\sum_{i=1}^{|\mathcal{F}_j|}\ell(x_i,y_i,\hat{y}_i) \quad (1)$$

where $\ell(\,)$ represents the classification loss incurred by the classification estimate $\hat{y}_i$[1] on the instance with covariates x_i and class y_i, then a "nearly unbiased" estimate of the classification loss under covariate shift can be computed as

$$\frac{1}{k}\sum_{j=1}^{k}\frac{1}{|\mathcal{F}_j|}\sum_{i=1}^{|\mathcal{F}_j|}\frac{P_{tst}(x_i)}{P_{tr}(x_i)}\ell(x_i,y_i,\hat{y}_i) \quad (2)$$

Here the term "nearly unbiased" means that the estimate becomes unbiased as the sample size $n\to\infty$. In the case of leave-one-out cross-validation, IWCV provides an unbiased estimate of the classification loss for a dataset with $n-1$ samples [51].

Under MAR bias, we have that $P_{tr}(x_i)=P(s=1|x_i)P_{tst}(x_i)$, meaning that $P_{tst}(x_i)/P_{tr}(x_i)=P(s=1|x_i)^{-1}$. Thus, "correcting" for MAR bias under simple loss functions amounts to estimating $P(s=1|x_i)$. This estimation can be accomplished in practice by building a classifier to predict $F:\mathbf{x}\to s$, that is, building a classifier with s as the class label. Such a construction is often feasible in practical applications. In credit scoring, for example, we only know the class label y (default) of applicants for whom $s=1$ (meaning credit was approved). However, creditors retain the application information for all applicants even those for whom $s=0$ (credit is denied) [5,61].

Effective correction of MAR bias, then, reduces to the problem of producing a *well-calibrated classifier* which predicts $P(s=1|x_i)$ as accurately as possible. In general this is not trivial, as many algorithms (such as Naive Bayes and Boosting) have been shown to produce probabilities that are skewed toward 0 or 1 [41,63].

Definition 8. *Missing not at random (MNAR)* occurs when there is no independence assumption between x, y and s. This kind of bias can introduce one or more of covariate shift, prior probability shift and concept shift.

Under MNAR bias, the selection mechanism may depend on the class attribute as well as the observed features. The most famous method for correcting MNAR bias comes from Heckman [28] who shows how to estimate a linear model over both observed and unobserved data when the dependent variable is known only for the observed data. Specifically, assume we have linear models for both the class variable y and the selection variable s of the form:

$$y_i=\beta_1 x_{1i}+u_{1i}$$

$$s_i=\beta_2 x_{2i}+u_{2i}$$

$$u1,u2\sim N(\mathbf{0},\sigma_{u1}^2,\rho) \quad (3)$$

Here the two β_j are 1-by-k_j model parameter vectors and the two x_{ji} are k_j-by-1 feature vectors for individual instances i. The vector x_{1i} contains the features upon which the class value depends, and x_{2i} contains the features on which the selection process depends. Thus, in Heckman's model, the class and selection variables are linear in some feature space with potentially correlated Gaussian noise.

Heckman proves that with these assumptions, an unbiased model y^* for the entire dataset can be built with the following procedure:

1. Estimate the parameters of the model s_i by some method such as ordinary least squares.
2. Set $\lambda_i=\phi(x_{2i}\beta_2)/\Phi(x_{2i}\beta_2)$.
3. Estimate the parameters of a new linear model y^* which includes λ as an independent variable.

Here ϕ and Φ are the standard normal PDF and CDF, respectively. Zadrozny and Elkan [62] generalize this procedure for arbitrary classification tasks by building one classifier to predict the selection label s and incorporating that classifier's predictions into a second classifier for predicting the class label y. While this approach has no theoretical guarantees, it was shown to be effective in a real-world application.

For completeness sake, we have defined a fourth option to be considered:

Definition 9. *Missing at random-class* (*MARC*) occurs when s depends on y but conditional on y is independent of x; so that $P(s=1|y,x)=P(s=1|y)$. This kind of bias can potentially produce prior probability shift.

Sufficient and necessary conditions for sample selection bias: Quiñonero-Candela et al. [44] give a set of conditions that the densities P_{tr} and P_{tst} need to satisfy in order for the classification problem to be modeled as a sample selection bias problem, meaning that its training and test densities can be expressed as in Definition 5. These conditions can be stated as follows:

1. *Support condition* $P_{tr}(x)>0\to P_{tst}(x)>0$.
2. *Selection condition* $\sup_x(P_{tr}(x,y)/P_{tst}(x,y)<\infty)$.

The support condition simply states that any feature vector x that can be drawn from the training distribution can also be drawn from the test distribution. The selection condition is slightly stronger, requiring that any pair (x,y) of a feature vector and class label that can be drawn from the $P_{tr}(x,y)$ can also be drawn from $P_{tst}(x,y)$. Fig. 6 explains this graphically. The red histogram shows a potential test density, the black histogram is a training density that may have been generated by sample selection bias (its density is nonzero everywhere the test density is nonzero) and the blue histogram shows a density that must be modeled by some other form of dataset shift.

This observation exposes a key difference between sample selection bias and covariate shift. Even in the case (MAR) where $p(s=1)$ depends only on the feature vector x, the framework of sample selection bias imposes a *stricter* criterion on the relationship between P_{tr} and P_{tst} than covariate shift. That is to say, there are some instances of covariate shift that cannot arise from MAR bias, but every instance of MAR bias can be modeled as covariate shift (Fig. 6(a)). As such, any technique that is developed to correct for covariate shift should also be able to correct for MAR bias, but the reverse is not true.

6.2. Challenges in correcting sample selection bias

We have seen that many established techniques to compensate for sample selection bias depend critically on the estimation of the selection variable s. In the case of IWCV, we need a well-calibrated estimate of $P(s=1|x)$ while the Zadrozny and Heckman techniques require a monotonic score. In either case if the chosen model is a poor fit, the correction procedure will be ineffective and may degrade rather than improve model performance [48].

If the feature sets x_1 and x_2 are identical (i.e. the same features are used to estimate both s and y), then the additional variable λ may end up highly correlated with the "uncorrected" estimate y_1. In this case, the Heckman procedure has little power to correct for sample selection bias. Little and Rubin [36] state that the Heckman procedure requires "significant" predictive variables in x_2 that are

[1] It is worth noting that the "classification estimate" may be real-valued, such as an estimate of $p(1|x)$.

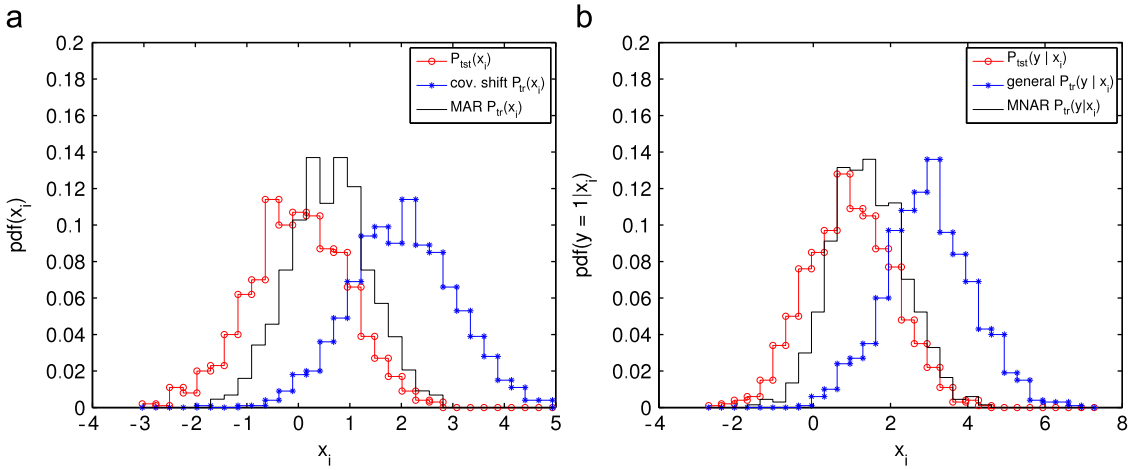


Fig. 6. Sufficient and necessary conditions for sample selection bias. The red curve shows a test pdf and the black and blue curves show potential training pdfs. The black density may be modeled as sample selection bias. The blue curve violates the (a) support condition and (b) selection condition. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

not in x_1 in order to be effective in many cases [43]. A broader survey of critiques to the Heckman correction can be found in [43].

When attempting to correct MAR bias with techniques such as importance-weighted cross-validation, one may run into trouble if $P(s=1|x_i)=0$. This situation, often referred to as *censorship*, arises when a deterministic procedure (such as a credit model) determines the value of s. Censorship may be addressed by modeling the problem as MNAR regardless of any explicit dependency on the class label y [12].

6.3. Non-stationary environments

In real-world applications, it is often the case that the data is not (time- or space-) stationary. Depending on the type of problem, non-stationary environments can introduce different kinds of shift:

- In $X \rightarrow Y$ problems, a non-stationary environment could create changes in either $P(x)$ or $P(y|x)$, generating covariate shift or concept shift, respectively.
- In $Y \rightarrow X$ problems, it could generate prior probability shift with a change in $P(y)$ or concept shift with a change in $P(x|y)$.

One of the most relevant non-stationary scenarios involves adversarial classification problems, such as spam filtering and network intrusion detection. This type of problem is receiving an increasing amount of attention in the machine learning field [16,6,10,35], and usually copes with non-stationary environments due to the existence of an adversary that tries to work around the existing classifier's learned concepts. In terms of the machine learning task, this adversary warps the test set so that it becomes different from the training set, thus introducing any possible kind of dataset shift.

There are also other applications where non-stationariness appears. They include remote sensing applications, where a dataset collected in a given season for an area with different terrains is employed to train the classifier but, when that classifier is deployed, mismatches may appear due to seasonal changes or because the new region has a different terrain distribution [3]; direct mail marketing, where the proportion of target customers or customer profiles may vary from one city to the next; and biometric authentication, among others.

7. Proposals in the literature for the analysis of dataset shift

In this section we give a brief overview of the different proposals that have appeared in the literature to work under the different types of dataset shift.

Covariate shift has been extensively studied in the literature, and a number of proposals to work under it have been published. Some of the most important ones include weighting the log-likelihood function [47], importance-weighted cross-validation [51], asymptotic Bayesian generalization error [59], discriminative learning [9], kernel mean matching [23], or adversarial search [22].

Prior probability shift has also been studied deeply, with a multitude of proposals appearing in the literature. There are two main strategies when designing classifiers for expected prior probability shift conditions:

- *Adaptive approaches*: These proposals train a classifier over the available data and then the adapt some of its parameters according to the (usually unlabeled) test data. This adaptation may be done either by the end user [33,31] or automatically [46,3].
- *Robust approaches*: Base the choice of classifier on some measure that is ideally transparent to changes in class distribution. The best known example would be ROC curve analysis [1,42] (which has generated some controversy, see [56,20]), but there are others too [18,2]. The automatic choice of classifier parameters [32] can also be considered a robust approach.

Other significant proposals in the literature have focused on determining the existence and/or shape of dataset shift between two datasets. Wang et al. [55] present the idea of correspondence tracing. They propose an algorithm for the discovering of changes of classification characteristics, which is based on the comparison between two rule-based classifiers, one built from each dataset. Yang et al. [60] present the idea of conceptual equivalence as a method for contrast mining, which consists of the discovery of discrepancies between datasets. Chawla and coworkers [14,45] developed a statistical framework to analyze changes in data distribution resulting in fractures between the data.

Lastly, there are some approaches that try and modify the data to repair dataset shift. Among them, Klinkenberg [32] proposed an example selection/weighting approach and Moreno-Torres et al. [40] applied a GP-based feature extraction technique to repair fractures between data originated in different biological laboratories by finding a transformation over the data from one of the laboratories.

8. Concluding remarks

In many practical applications of machine learning, the data available for model-building (training data) are not strictly representative of the data on which the classifier will ultimately be deployed (test data). This problem, which we call *dataset shift* in accordance with [44] generalizes a wide variety of researches that are scattered throughout the machine learning literature. The purpose of this paper is to survey and unify this research in order to better inform future endeavors in the field.

Researchers studying the general problem of dataset shift, or specific instances of this problem, have coined a number of different names for it. These include *concept shift* [57], *concept drift* [57], *covariate shift* [47], *data fracture* [14,40], *reject inference* [24,15], and *imprecise class distributions* [2], among others. Worse still, researchers have sometimes used different terms to refer to the same problem, or given different definitions to the same term. To clear up this confusion and to make future research easier, we have carefully studied the terminology used in the literature and proposed a common convention which attempts to capture the essence of the terms as they are most commonly used. Specifically, we propose:

- *Covariate shift* if $P_{tst}(x) \neq P_{tr}(x)$ but $P_{tst}(y|x) = P_{tr}(y|x)$, in accordance with [47].
- *Prior probability shift* if $P_{tst}(y) \neq P_{tr}(y)$ but $P_{tst}(y|x) = P_{tr}(y|x)$.
- *Concept shift* if $P_{tst}(x) = P_{tr}(x)$ but $P_{tst}(y|x) \neq P_{tr}(y|x)$ (in $X \rightarrow Y$ problems) or $P_{tst}(x|y) \neq P_{tr}(x|y)$ (in $Y \rightarrow X$ problems).
- *Dataset shift* if $P_{tst}(x,y) \neq P_{tr}(x,y)$ but none of the above hold.

Next, we survey common causes of dataset shift. *Sample selection bias* [12,28,61] occurs when the training sample is selected non-uniformly at random from the test population. Depending on the selection criteria and the type of classification problem, selection bias may produce covariate shift, prior probability shift, or general dataset shift. In *adversarial* environments [10,16,35] such as spam detection and fraud detection, *adversaries* continually adapt the test data to the output of the classification algorithm. The adversaries try to produce data (with some constraints) which the learner will misclassify as often as possible. This tends to produce general dataset shift as the adversary may alter the test distribution arbitrarily. In *non-stationary* environments, the dataset shift arises from a significant physical or temporal difference between training and test data sources. If a model trained on one continent is applied on another, for example, arbitrary changes in data distribution may result.

Finally, we have briefly surveyed some proposals in the literature for learning under dataset shift, either *detecting* that a shift has occurred or *adapting* to the shift once it does occur. We plan to expand on this in much greater detail in future work.

Acknowledgments

Jose García Moreno-Torres is currently supported by an FPU Grant from the Ministerio de Educación y Ciencia of the Spanish Government. This work was supported in part by the Spanish Government's KEEL project (TIN2008-06681-C06-01). This work was also supported in part by the National Science Foundation (NSF) Grant ECCS-0926170. Lastly, the work was also partially supported by the Spanish projects DPI2009-08424 and TEC2008-01348/TEC.

References

[1] N.M. Adams, D.J. Hand, Comparing classifiers when the misallocation costs are uncertain, Pattern Recognition 32 (7) (1999) 1139–1147.

[2] R. Alaiz-Rodríguez, A. Guerrero-Curieses, J. Cid-Sueiro, Minimax regret classifier for imprecise class distributions, Journal of Machine Learning Research 8 (2007) 103–130.

[3] R. Alaiz-Rodríguez, A. Guerrero-Curieses, J. Cid-Sueiro, Classification under changes in class and within-class distributions, in: Proceedings of the 10th International Work-Conference on Artificial Neural Networks, IWANN '09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 122–130.

[4] R. Alaiz-Rodríguez, N. Japkowicz, Assessing the impact of changing environments on classifier performance, in: Proceedings of the Canadian Society for Computational Studies of Intelligence, 21st Conference on Advances in Artificial Intelligence, Canadian AI '08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 13–24.

[5] J. Banasik, J. Crook, L. Thomas, Sample selection bias in credit scoring models, Journal of the Operational Research Society 54 (8) (2003) 822–832.

[6] M. Barreno, B. Nelson, A.D. Joseph, J.D. Tygar, The security of machine learning, Machine Learning (2010) 121–148.

[7] M. Basu, T.K. Ho, Data Complexity in Pattern Recognition, Springer-Verlag Inc., New York, Secaucus, NJ, USA, 2006.

[8] S. Bickel, M. Brückner, T. Scheffer, Discriminative learning for differing training and test distributions, in: Proceedings of the 24th International Conference on Machine Learning, ICML 2007, ACM, New York, NY, USA, 2007, pp. 81–88.

[9] S. Bickel, M. Brückner, T. Scheffer, Discriminative learning under covariate shift, Journal of Machine Learning Research 10 (2009) 2137–2155.

[10] B. Biggio, G. Fumera, F. Roli, Multiple classifier systems for robust classifier design in adversarial environments, International Journal of Machine Learning and Cybernetics 1 (2010) 27–41.

[11] C.E. Brodley, P. Uiversity, M.A. Friedl, B. Uiversity, B.P. Edu, Identifying mislabeled training data, Journal of Artificial Intelligence Research 11 (1999) 131–167.

[12] N. Chawla, G. Karakoulas, Learning from labeled and unlabeled data: an empirical study across techniques and domains, Journal of Artificial Intelligence Research 23 (1) (2005) 331–366.

[14] D.A. Cieslak, N.V. Chawla, A framework for monitoring classifiers' performance: when and why failure occurs? Knowledge and Information Systems 18 (1) (2009) 83–108.

[15] J. Crook, J. Banasik, Does reject inference really improve the performance of application scoring models? Journal of Banking & Finance 28 (4) (2004) 857–874.

[16] N. Dalvi, P. Domingos, Mausam, S. Sanghai, D. Verma, Adversarial classification, in: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, ACM, New York, NY, USA, 2004, pp. 99–108.

[17] T.G. Dietterich, G. Widmer, M. Kubat, Special issue on context sensitivity and concept drift, Machine Learning 32 (2) (1998).

[18] C. Drummond, R.C. Holte, Explicitly representing expected cost: an alternative to ROC representation, in: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, pp. 198–207.

[19] A. Farhangfar, L. Kurgan, J. Dy, Impact of imputation of missing values on classification error for discrete data, Pattern Recognition 41 (12) (2008) 3692–3705.

[20] T. Fawcett, P.A. Flach, A response to Webb and Ting's 'on the application of ROC analysis to predict classification performance under varying class distributions', Machine Learning 58 (1) (2005) 33–38.

[21] M. Ghannad-Rezaie, H. Soltanian-Zadeh, H. Ying, M. Dong, Selection–fusion approach for classification of datasets with missing values, Pattern Recognition 43 (6) (2010) 2340–2350.

[22] A. Globerson, C.H. Teo, A. Smola, S. Roweis, An adversarial view of covariate shift and a minimax approach, in: J. Quiñonero Candela, M. Sugiyama, A. Schwaighofer, N.D. Lawrence (Eds.), Dataset Shift in Machine Learning, The MIT Press, 2009, pp. 179–198.

[23] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, B. Schölkopf, Covariate shift by kernel mean matching, in: J. Quiñonero Candela, M. Sugiyama, A. Schwaighofer, N.D. Lawrence (Eds.), Dataset Shift in Machine Learning, The MIT Press, 2009, pp. 131–160.

[24] D. Hand, Reject inference in credit operations, in: Credit risk modeling: design and application, 1998, pp. 181–190.

[25] D. Hand, W. Henley, Statistical classification methods in consumer credit scoring: a review, Journal of the Royal Statistical Society: Series A 160 (3) (1997) 523–541.

[26] D.J. Hand, Rejoinder: classifier technology and the illusion of progress, Statistical Science 21 (1) (2006) 30–34.

[27] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.

[28] J. Heckman, Sample selection bias as a specification error, Econometrica: Journal of the Econometric Society (1979) 153–161.

[29] T.K. Ho, M. Basu, Complexity measures of supervised classification problems, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (3) (2002) 289–300.

[30] J. Huang, A.J. Smola, A. Gretton, K.M. Borgwardt, B. Schölkopf, Correcting sample selection bias by unlabeled data, Advances in Neural Information Processing Systems 19 (2007) 601–608.

[31] M.G. Kelly, D.J. Hand, N.M. Adams, The impact of changing populations on classifier performance, in: Proceedings of the Fifth ACM SIGKDD International

Conference on Knowledge Discovery and Data Mining, KDD 99, 1999, pp. 367–371.
[32] R. Klinkenberg, Learning drifting concepts: example selection vs. example weighting, Intelligent Data Analysis 8 (3) (2004) 281–300.
[33] M. Kubat, R.C. Holte, S. Matwin, Machine learning for the detection of oil spills in satellite radar images, Machine Learning 30 (2–3) (1998) 195–215.
[34] T. Lane, C.E. Brodley, Approaches to online learning and concept drift for user identification in computer security, in: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, KDD, AAAI Press, 1998, pp. 259–263.
[35] P. Laskov, R. Lippmann, Machine learning in adversarial environments, Machine Learning 81 (2010) 115–119.
[36] R. Little, D. Rubin, Statistical Analysis with Missing Data, 1987.
[37] R.J.A. Little, D.B. Rubin, Statistical Analysis with Missing Data, Probability and Statistics, second ed., Wiley, New Jersey, 2002.
[38] Z.-Y. Liu, H. Qiao, Multiple ellipses detection in noisy environments: a hierarchical approach, Pattern Recognition 42 (11) (2009) 2421–2433.
[39] J. Luengo, S. García, F. Herrera, A study on the use of imputation methods for experimentation with Radial Basis Function Network classifiers handling missing attribute values: the good synergy between rbfns and eventcovering method, Neural Networks 23 (3) (2010) 406–418.
[40] J.G. Moreno-Torres, X. Llorà, D.E. Goldberg, R. Bhargava, Repairing fractures between data using genetic programming-based feature extraction: a case study in cancer diagnosis, Information Sciences, in press, doi:10.1016/j.ins.2010.09.018.
[41] A. Niculescu-Mizil, R. Caruana, Predicting good probabilities with supervised learning, in: Proceedings of the ICML, ACM, 2005, pp. 625–632.
[42] F. Provost, T. Fawcett, Robust classification for imprecise environments, Machine Learning 42 (3) (2001) 203–231.
[43] P. Puhani, The Heckman correction for sample selection and its critique, Journal of Economic Surveys 14 (1) (2000) 53–68.
[44] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, N.D. Lawrence, Dataset Shift in Machine Learning, The MIT Press, 2009.
[45] T. Raeder, N.V. Chawla, Model monitor: evaluating, comparing, and monitoring models, Journal of Machine Learning Research 10 (2009) 1387–1390.
[46] M. Saerens, P. Latinne, C. Decaestecker, Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure, Neural Computation 14 (1) (2002) 21–41.
[47] H. Shimodaira, Improving predictive inference under covariate shift by weighting the log-likelihood function, Journal of Statistical Planning and Inference 90 (2) (2000) 227–244.
[48] R. Stolzenberg, D. Relles, Tools for intuition about sample selection bias and its correction, American Sociological Review 62 (3) (1997) 494–507.
[49] A. Storkey, When training and test sets are different: characterizing learning transfer, in: J. Quiñonero Candela, M. Sugiyama, A. Schwaighofer, N.D. Lawrence (Eds.), Dataset Shift in Machine Learning, The MIT Press, 2009, pp. 3–28.
[51] M. Sugiyama, M. Krauledat, K.-R. Müller, Covariate shift adaptation by importance weighted cross validation, Journal of Machine Learning Research 8 (2007) 985–1005.
[52] Y. Sun, M.S. Kamel, A.K. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recognition 40 (12) (2007) 3358–3378.
[53] Y.M. Sun, A.K.C. Wong, M.S. Kamel, Classification of imbalanced data: a review, International Journal of Pattern Recognition and Artificial Intelligence 23 (4) (2009) 687–719.
[54] K.M. Ting, A study on the effect of class distribution using cost-sensitive learning, in: Fifth International Conference on Discovery Science, DS 2002, 2002, pp. 98–112.
[55] K. Wang, S. Zhou, C.A. Fu, J.X. Yu, F. Jeffrey, X. Yu, Mining changes of classification by correspondence tracing, in: Proceedings of the 2003 SIAM International Conference on Data Mining (SDM 2003), 2003, pp. 95–106.
[56] G.I. Webb, K.M. Ting, On the application of ROC analysis to predict classification performance under varying class distributions, Machine Learning 58 (1) (2005) 25–32.
[57] G. Widmer, M. Kubat, Learning in the presence of concept drift and hidden contexts, Machine Learning 23 (1996) 69–101.
[58] X. Wu, X. Zhu, Mining with noise knowledge: error aware data mining, IEEE Transactions on SMC, Part A 28 (4) (2008) 917–932.
[59] K. Yamazaki, M. Kawanabe, S. Watanabe, M. Sugiyama, K.-R. Müller, Asymptotic Bayesian generalization error when training and test distributions are different, in: Proceedings of the 24th International Conference on Machine Learning, ICML '07, ACM, New York, NY, USA, 2007, pp. 1079–1086.
[60] Y. Yang, X. Wu, X. Zhu, Conceptual equivalence for contrast mining in classification learning, Data & Knowledge Engineering 67 (3) (2008) 413–429.
[61] B. Zadrozny, Learning and evaluating classifiers under sample selection bias, in: Proceedings of the 21st International Conference on Machine Learning, ICML '04, ACM, New York, NY, USA, 2004, p. 114.
[62] B. Zadrozny, C. Elkan, Learning and making decisions when costs and probabilities are both unknown, in: Proceedings of the KDD, ACM, 2001, pp. 204–213.
[63] B. Zadrozny, C. Elkan, Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers, in: Proceedings of the ICML, 2001, pp. 609–616.
[64] X. Zhu, X. Wu, Class noise vs attribute noise: a quantitative study, Artificial Intelligence Review 22 (3) (2004) 177–210.

Jose G. Moreno-Torres received the M.Sc. degree in Computer Science in 2008 from the University of Granada, Spain. After spending a year as a fellow of an international "la Caixa" scholarship, during which he did research at the IlliGAL laboratory under the supervision of Prof. David E. Goldberg, he is currently a Ph.D. candidate under the supervision of Prof. Francisco Herrera, working with the Soft Computing and Intelligent Information Systems Group in the Department of Computer Science and Artificial Intelligence at the University of Granada. His current research interests include dataset shift, imbalanced classification, bibliometrics and multi-instance learning.

Troy Raeder is a Ph.D. student in Computer Science at the University of Notre Dame in South Bend, IN, USA. His research interests include scenario analysis in machine learning, evaluation methodologies in machine learning, and robust models for changing data distributions. He received B.S. and M.S. degrees in Computer Science from Notre Dame in 2005 and 2009, respectively.

Rocío Alaiz-Rodríguez received the B.S. degree in Electrical Engineering from the University of Valladolid, Spain, in 1999 and the Ph.D. degree from Carlos III University of Madrid, Spain. She is currently an Associate Professor at the Department of Electrical and Systems Engineering, University of Leon, Spain. Her research interests include learning theory, statistical pattern recognition, neural networks and their applications to image processing and quality assessment (in particular, food and frozen–thawed animal semen).

Nitesh V. Chawla is an Associate Professor in the Department of Computer Science and Engineering at the University of Notre Dame. He directs the Data Inference Analysis and Learning Lab (DIAL) and is also the co-director of the Interdisciplinary Center of the Network Science and Applications (iCenSA) at Notre Dame. His research is supported with research grants from organizations such as the National Science Foundation, the National Institute of Justice, the Army Research Labs, and Industry Sponsors. His research group has received numerous honors, including best papers, outstanding dissertation, and a variety of fellowships. He has also been noted for his teaching accomplishments, receiving the National Academy of Engineers CASEE New Faculty Fellowship, and the Outstanding Undergraduate Teacher Award in 2008 and 2011. He is an Associated Editor for IEEE Transactions of Systems, Man and Cybernetics Part B and Pattern Recognition Letters. More information is available at http://www.nd.edu/~nchawla.

Francisco Herrera received his M.Sc. degree in Mathematics in 1988 and Ph.D. degree in Mathematics in 1991, both from the University of Granada, Spain. He is currently a Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada. He has had more than 200 papers published in international journals. He is coauthor of the book "Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases" (World Scientific, 2001). He currently acts as Editor in Chief of the international journal "Progress in Artificial Intelligence" (Springer) and serves as Area Editor of the Journal Soft Computing (area of evolutionary and bioinspired algorithms) and International Journal of Computational Intelligence Systems (area of information systems). He acts as Associated Editor of the journals: IEEE Transactions on Fuzzy Systems, Information Sciences, Advances in Fuzzy Systems, and International Journal of Applied Metaheuristics Computing; and he serves as a member of several journal editorial boards, among others: Fuzzy Sets and Systems, Applied Intelligence, Knowledge and Information Systems, Information Fusion, Evolutionary Intelligence, International Journal of Hybrid Intelligent Systems, Memetic Computation, Swarm and Evolutionary Computation. He received the following honors and awards: ECCAI Fellow 2009, 2010 Spanish National Award on Computer Science ARITMEL to the "Spanish Engineer on Computer Science", and International Cajastur "Mamdani" Prize for Soft Computing (Fourth Edition, 2010). His current research interests include computing with words and decision-making, data mining, bibliometrics, data preparation, instance selection, fuzzy rule-based systems, genetic fuzzy systems, knowledge extraction based on evolutionary algorithms, memetic algorithms and genetic algorithms.

1.2. Tackling Dataset Shift in Classification: Benchmark and Methods

- J.G. Moreno-Torres, T. Raeder, R. Aláiz-Rodríguez, N.V. Chawla, F. Herrera, Tackling Dataset Shift in Classification: Benchmark and Methods. Submitted to IEEE Transactions on Neural Networks and Learning Systems.

  - Status: **Submitted**.
  - Impact Factor (JCR 2011): 2.952.
  - Subject Category: Computer Science, Artificial Intelligence. Ranking 12 / 111 (Q1).
  - Subject Category: Computer Science, Hardware & Architecture. Ranking 1 / 50 (Q1).
  - Subject Category: Computer Science, Theory & Methods. Ranking 4 / 99 (Q1).
  - Subject Category: Engineering, Electrical & Electronic. Ranking 19 / 245 (Q1).

Tackling Dataset Shift in Classification: Benchmark and Methods

Journal:	*IEEE Transactions on Neural Networks and Learning Systems*
Manuscript ID:	Draft
Manuscript Type:	Brief Paper
Date Submitted by the Author:	n/a
Complete List of Authors:	Moreno-Torres, Jose; University of Granada, Dept. Computer Science and Artificial Intelligence Raeder, Troy; University of Notre Dame, Computer Science and Engineering Aláiz-Rodríguez, Rocío; Universidad de León, Dpto. de Energía Eléctrica y de Sistemas Chawla, Nitesh; University of Notre Dame, Computer Science and Engineering Herrera, Francisco; University of Granada, Dept. Computer Science and Artificial Intelligence
Keywords:	dataset shift, data fracture, changing environments, differing training and test populations, covariate shift, sample selection bias, experimental review

SCHOLARONE™
Manuscripts

Tackling Dataset Shift in Classification: Benchmark and Methods

Jose G. Moreno-Torres*, Troy Raeder†, Rocío Aláiz-Rodríguez‡, Nitesh V. Chawla†, Francisco Herrera†
*Department of Computer Science and Artificial Intelligence, Universidad de Granada, 18071 Granada, Spain.
{jose.garcia.mt, herrera}@decsai.ugr.es
† Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA.
traeder@gmail.com, nchawla@cse.nd.edu
‡ Universidad de León, Dpto. de Ingeniería Eléctrica y de Sistemas,Campus de Vegazana, 24071 León, Spain.
rocio.alaiz@unileon.es

For Peer Review

***Abstract*—Dataset shift is a phenomenon that occurs in some degree in most real-world classification problems. It is defined as the situation where the data used to train the classifier and the environment where said classifier is deployed do not follow the same distribution. The impact this issue has on classification is similar to the existence of noisy examples [1], and the solutions applied have a lot in common with the field of transfer learning [2].**

The field of dataset shift has received a growing amount of interest in the last few years. The fact that most real-world applications have to cope with some form of shift makes its study highly relevant. This work presents a new set of datasets as a benchmark that allows a fair study of new proposals, along with detailed analysis of the most important algorithms on the field and a comparison them in terms of their effectiveness when dealing with a varied repertoire of datasets and shifts.

***Index Terms*—dataset shift , data fracture , changing environments , differing training and test populations , covariate shift , sample selection bias , experimental review**

I. Introduction

Dataset shift is defined as the situation where the data used to train the classifier and the environment where said classifier is deployed do not follow the same distribution. It is a common phenomenon in the field of classification, one that appears often on real-world applications and can have a relevant impact on classifier performance [3], [4]. The impact this issue has on classification is similar to the existence of noisy examples [5], and the solutions applied have a lot in common with the field of transfer learning [2].

There are a number of proposals in the literature, both for the detection of dataset shift and for its repair; but most of the proposed methods have not been compared against each other, resulting in a difficulty to measure the effectiveness of a new method. It is also unclear what impact does dataset shift have on classical classifier performances, and whether some of them are better suited to deal with it naturally than others.

This work presents an empirical analysis of classification algorithms on a number of dataset shift scenarios induced in different datasets. It includes a study of the impact of dataset shift over the performance of some representative classical classification methods. It also presents a detailed analysis on the performance of some of the most influential dataset shift algorithms in the literature over said datasets, presenting the first review with an experimental analysis ever done on the topic.

The rest of the paper is organized as follows: Section II provides the background and notations. Section III details the procedure employed to create the datasets we present; paying special attention to the different kinds of dataset shift sources that were applied. Section IV introduces the methods obtained from the literature that were tested in this paper. Section V presents the performance of the previously presented methods over the datasets shown. Finally, some concluding remarks are made in Section VI.

The website associated to this work is http://sci2s.ugr.es/dataset-shift, and it includes the benchmark datasets created for this study, the code of all the algorithms analyzed and the detailed results obtained by the experiments shown in this work.

II. Background

The first consideration when studying dataset shift is the characterizations of the different types of shifts that can appear and how these are generated. We include here a brief summary, for more details see [4]. For the definitions used in this section, assume X represents the covariates of the problem and Y the class label. We also use the problem categorization proposed in [6], according to which there are two kinds of problems:

- X→Y problems, where the class label is causally determined by the values of the covariates. A typical example would be credit card fraud detection, since the behavior of the user, represented in the covariate space X, determines the class label: whether there is fraud or not.
- Y→X problems, where the class label causally determines the values of the covariates. Medical diagnosis usually falls in this category, where the disease, which is modeled as the class label Y, determines the symptoms, represented in the machine learning task as covariates X.

There are four different types of shift, depending on which probabilities change or not:

- **Covariate shift** appears only in X→Y problems, and is defined as the case where $P_{tr}(y|x) = P_{tst}(y|x)$ and $P_{tr}(x) \neq P_{tst}(x)$.

- **Prior probability shift** appears only in Y→X problems, and is defined as the case where $P_{tr}(x|y) = P_{tst}(x|y)$ and $P_{tr}(y) \neq P_{tst}(y)$.
- **Concept shift** is defined as
  - $P_{tr}(y|x) \neq P_{tst}(y|x)$ and $P_{tr}(x) = P_{tst}(x)$ in X→Y problems.
  - $P_{tr}(x|y) \neq P_{tst}(x|y)$ and $P_{tr}(y) = P_{tst}(y)$ in Y→X problems.

Some of the most common causes for the appearance of dataset shift are the different types of sample selection bias, and also domain shift. They are defined as:

- **Sample Selection Bias: Missing at Random (MAR).** MAR occurs when the probability of sampling an example (that is, including it in the training set) depends on x, but is independent on y.
- **Sample Selection Bias: Missing Not at Random (MNAR).** MNAR is the case where the sampling probability is both dependent on x and y.
- **Sample Selection Bias: Missing at Random - Class (MARC).** MARC appears when the sampling probability depends exclusively on y.
- **Domain Shift (DS).** This shift appears when there is a change in the scale of one or more of the attributes in x.

While sample selection bias and domain shift can be considered the most typical sources of dataset shift in real-world problems, a number of other causes exist, such as source component shift [7](a class is composed of several subclasses, and the prior probabilities of said subclasses change), adversarial environments [8](for instance, a hacker trying to break a security measure will try to disguise himself to be as different as possible from previous hackers), imbalanced classification problems [9] (where rare examples or small disjuncts greatly increase the impact of even very soft degrees of dataset shift), or even dataset shift artificially introduced in a cross-validation setup [10].

On this work we choose to focus only on sample selection bias and domain shift, since the others are much harder to characterize and thus to recreate artificially to obtain a battery of datasets suitable for experimentation.

III. Benchmark: Datasets and Shifts

To create a benchmark for the analysis and comparison of the different types of solutions to dataset shift proposed in the literature, different shifts have been artificially introduced over well-known real datasets. The datasets employed are presented in subsection III-A, while a detailed explanation of the procedure used to introduce shifts is shown in subsection III-B.

A. Datasets

A list of the original datasets used to produce all the shifted datasets is presented here in Table I. Note that only datasets with numeric attributes were used since most of the methods in the literature are unable to deal with nominal attributes; and also that samples with missing values (which appeared in some of these original datasets) were eliminated for the same reason. These datasets were obtained from the KEEL dataset repository [11].

TABLE I
Datasets used in the study as the basis to produce dataset shift problems

Dataset	# attributes	# instances	# classes
appendicitis	8	106	2
breast-w	10	683	2
bupa	7	345	2
ion	35	351	2
heart	14	270	2
pima	9	768	2
ringnorm	21	300	2
sonar	61	208	2
threenorm	21	300	2

Note that the number of instances shown takes into account the elimination of instances with missing values.

B. Introduced shifts

Since the goal of this work is to simulate real-world conditions to check the effectiveness of a number of methods, we decided not to introduce arbitrary shifts but rather to apply different potential sources of dataset shift to the data. Note that the procedure to generate the shifted datasets is slightly different when working with sample selection bias than when dealing with the other types of shifts. In the sample selection bias case, the dataset is partitioned and then each example in the training set is run through the bias to decide whether it is included in the final training set. On the other hand, when introducing domain shift; the shift is applied to all the examples in the test set. The four sources of dataset shift employed in this study are:

- MAR is implemented by having three different types of MAR: Top%, Gaussian and an Interval-based function, applied over some attribute in X, that would decide whether an example gets included in the final training set or not.
- MNAR is implemented in the following way: For all examples of the positive class, introduce a MAR bias; for all examples of the negative class, introduce a different MAR bias.
- MARC is implemented by assigning a different selection probability to examples of the positive class than those of the negative class.
- Domain shift is implemented as a linear rescaling of a single attribute in X.

Each of the above sources of dataset shift has a number of parameters, which can be seen in Table II. The range column represents the potential values the parameter was allowed to take when being decided randomly.

To provide meaningful results and achieve as high experimental reliability as possible, the following consideratios were followed to generate the datasets used in this study:

- Four degrees of shift were defined as shown in Table III.
- We chose to employ rather extreme ranges to test the capabilities of the repairing methods in the literature.

[1]Several intervals are generated, for example $[0.2, 0.3] \cup [0.56, 0.73]$

For Peer Review

TABLE II
PARAMETERS NEEDED BY EACH DATASET SHIFT GENERATOR. X IS THE ATTRIBUTE ALONG WHICH THE SHIFT IS INJECTED.

Source of Dataset Shift	Parameter	Range	Description
Gaussian MAR	mean	$[min(x), max(x)]$	Mean of the normal dist.
	std	$[0.1 * (max(x) - min(x)), 0.2 * (max(x) - min(x))]$	Standard deviation of the normal distribution
Interval MAR	Interval	$[min(x), max(x)]$ [1]	Accept examples in the interval, reject the rest
TopN% MAR	N	$[0, 100]$	Accept the top N% examples, ranked according to attribute x
MNAR	biasPos	Any MAR Bias	Apply to negative class
	biasNeg	Any MAR Bias	Apply to positive class
MARC	p0	$[0, 1]$	$p(s\|\|y = 0)$
	p1	$[0, 1]$	$p(s\|\|y = 1)$
Domain Shift	mult	$[0.1, 10]$	$f(x) = x * mult + add$
	add	$[-mult * (max(x) - min(x))/2, mult * (max(x) - min(x))/2]$	

- In order to obtain a significant sample of datasets, we applied each source of Dataset Shift shown above to create 25 instances of each dataset included in the study for each degree of shift.

TABLE III
DEGREES OF SHIFT

Degree of shift	Min. % of examples selected	Max. % of examples selected
Low	70	98
Medium	40	60
High	25	37
Extreme	10	20

The actual procedure to generate all the datasets is as follows:

1 For each dataset in Table I, create 100 random different partitionings (using the method described in [10] to avoid introducing extra, unaccounted for, dataset shift) for 5-fold cross validation.
2 Save those 100 partitions as data with no shift.
3 Assign 25 of those partitions to each degree of shift, and apply all the dataset shift sources described in Table II to obtain shifted datasets. This generates $25 * 6 = 150$ datasets for each degree of shift, each with its corresponding 5-fold partitioning.
4 In the end, for each original dataset, we have:
   - 100 non-biased datasets.
   - 150 low-shift datasets, 25 created with each source of shift.
   - 150 medium-shift datasets, 25 created with each source of shift.
   - 150 high-shift datasets, 25 created with each source of shift.
   - 150 extreme-shift datasets, 25 created with each source of shift.

Note that, when we show results for MAR bias, it corresponds to the average results obtained from Gaussian, Interval and TopN MAR biases.

The 700 partitions for each dataset can be found in the associated website http://sci2s.ugr.es/dataset-shift in arff format, available for download.

IV. ALGORITHMS FOR DATASET SHIFT STUDIED

This section presents the methods chosen from the literature to be compared in this study. They are classified according to their specialization. We organize the section according to the following three subsections: IV-A describes the classical classifiers used to test the impact of dataset shift using no further solution, subsection IV-B is dedicated to the methods specialized in working under covariate shift and, lastly, subsection IV-C describes the remaining methods that focus on solving other types of dataset shift, or that are designed as general dataset shift solvers.

Regarding the algorithms mentioned in subsections IV-B and IV-C; we present here only a short description, more details can be found in http://sci2s.ugr.es/dataset-shift.

A. Classical Classification Methods

This subsection briefly describes the well-studied classifiers in the literature that were chosen to check the impact of dataset shift on classifier performance. They were picked taking into account the fact that the shift introduction procedure works along a single attribute, thus avoiding classifiers that could ignore it while at the same time trying to maximize diversity, and with the goal of checking how their performances degrade in the presence of increasingly acute dataset shift. However, it should be noted that this is not a study dedicated to comparing cutting-edge classifiers, and thus only the basic versions of each family of methods were included. They can be seen in Table IV.

TABLE IV
CLASSIFICATION ALGORITHMS USED

Algorithm	Abbreviation	Type of classifier
Nearest neighbor k=1 [12]	1NN	Lazy learner
C4.5 [13]	C45	Decision tree
Sequential minimal optimization [14]	SMO	Support vector machine

B. Covariate Shift Solvers

The algorithms in this subsection are designed to cope with covariate shift [15]:

For Peer Review

- **Importance Weighted Cross Validation (IWCV)** [16], [17]. Applies different weights to each sample to balance out data distribution in the presence of covariate shift.
- **Integrated Optimization Problem (IOP)** [18]. Treats the learning under covariate shift as an integrated optimization problem, whose instantiation leads to a kernel logistic regression and an exponential model classifier for covariate shift.
- **Kernel Mean Matching (KMM)** [19]. Reweighs training data to even the distribution with test data by matching covariate distributions in a high dimensional space.

C. General Dataset Shift Solvers

This subsection includes the remaining solvers, that are not specialized on covariate shift but rather tackle the problem from a more general point of view.

- **SUbclass RE-estimation (SCRE)** [20]. This method, which does not require labels for the test set, was designed to tackle source component shift; but is also capable of dealing with other types. The idea behind it is to reestimate prior probabilities based on the different subclass distribution of the test set; which is obtained by the application of a clustering method.
- **GP-RFD** [21]. Requires the test set to be partially labeled. Uses the performance of a classifier built on the training set over the labeled examples of the test to drive a Genetic-Programming based evolution that designs a transformation of the test set into a new one where the old classifier (the one that was built over the training set) has the best possible performance.

A note on the parametrization of the methods: default parameters recommended by each author, except for the GP-RFD method, where the population size was set to 100, and the number of generations to 50.

V. Experimental results

This section first presents the classifier performance of the classical classifier methods in subsection V-A, followed by the results obtained by the dataset shift solvers in subsection V-B. Due to space concerns, only a summary of the results is shown here, a more complete version is available at http://sci2s.ugr.es/dataset-shift.

A. Classical classification methods

Here we present the results obtained by the classifiers mentioned in section IV-A for the different types of shift and the different degrees of shift explained before. The results are grouped by type of shift in Table V. Only the test set performance (as mentioned earlier, with 5-fold cross validation) is shown here, since dataset shift does not affect training set performance particularly.

Some conclusions can be extracted by looking at this data:

- MAR seems to affect all classifiers very similarly. A low degree of shift has very little impact, but starting from medium shift a significant performance decrease is noticed.

TABLE V
Impact of dataset shift on classical classifiers. Average test set performance.

Problem / Classifier	C45	1NN	SMO
No Shift	0.7728	0.7530	0.7723
MAR: Low Shift	0.7575	0.7423	0.7578
MAR: Medium Shift	0.7093	0.6983	0.7140
MAR: High Shift	0.6844	0.6808	0.6862
MAR: Extreme Shift	0.6134	0.6332	0.6222
MNAR: Low Shift	0.7574	0.7496	0.7718
MNAR: Medium Shift	0.6935	0.7246	0.7299
MNAR: High Shift	0.6482	0.7005	0.7018
MNAR: Extreme Shift	0.5819	0.6526	0.6350
MARC: Low Shift	0.7374	0.7224	0.7340
MARC: Medium Shift	0.7007	0.6888	0.6780
MARC: High Shift	0.6686	0.6792	0.6456
MARC: Extreme Shift	0.6075	0.6352	0.5674
DS: Low Shift	0.7503	0.7483	0.7616
DS: Medium Shift	0.7561	0.7478	0.7636
DS: High Shift	0.7551	0.7430	0.7500
DS: Extreme Shift	0.7482	0.7467	0.6813

- MNAR behaves very similarly to MAR.
- MARC produces an effect that is also very similar to the previous ones, with a first drop in performance at medium shift and a rapid degradation from there on.
- Domain shift affects SMO and pretty relevantly, but both C45 and 1NN seem capable of coping with it without losing much performance.
- In summary, these results show that low degrees of shift are not particularly worrisome for the classifiers tested, but that significant performance is lost when at least a medium degree of sample selection bias is present for all classifiers, and that domain shift also affects SMO classifiers.

B. Dataset shift solvers

In this section, the test performance of 5 dataset shift solvers is compared over the previously mentioned classifiers. Note that three of the methods are specifically designed to cope with covariate shift (IWCV, IOP and KMM), and two others are general shift solvers (GP-RFD and SCRE). It is also important to keep in mind that some of these methods do not require any labels from the test set, but that GP-RFD does (needs a partially labeled test set), so this experiment intends not to be a fair comparison between methods in equal terms, but rather an overview of where the state of the art is at in terms of efficiency when working under dataset shift conditions.

Table VI shows the average classification test results obtained by all dataset shift solvers. In the case of GP-RFD, which acts as a preprocessing method, a C4.5 classifier was applied. C4.5 was chosen as it is in the middle range of classifiers in terms of dataset shift sensibility (see Table V), and is a well understood method that is not particularly sensitive to parametrization. The 'No solver' column corresponds to the performance of the C4.5 classifier without any dataset solver, same data that was shown in Table V.

For Peer Review

TABLE VI
AVERAGE TEST AUC OBTAINED AFTER APPLYING DATASET SHIFT SOLVERS

Problem / Solver	None	IWCV	IOP	KMM	SCRE	GP-RFD
No Shift	0.7728	0.7206	0.7089	0.6596	0.8455	0.6470
MAR: Low Shift	0.7575	0.6929	0.6829	0.6386	0.8215	0.6468
MAR: Medium Shift	0.7093	0.6669	0.6602	0.6188	0.7430	0.6313
MAR: High Shift	0.6844	0.6565	0.6291	0.6198	0.7121	0.6284
MAR: Extreme Shift	0.6134	0.5881	0.6089	0.6201	0.5675	0.6058
MNAR: Low Shift	0.7574	0.7092	0.6999	0.6595	0.8384	0.6441
MNAR: Medium Shift	0.6935	0.6747	0.6699	0.6397	0.7925	0.6385
MNAR: High Shift	0.6482	0.6574	0.6585	0.6176	0.7620	0.6360
MNAR: Extreme Shift	0.5819	0.6046	0.6156	0.5994	0.6895	0.6225
MARC: Low Shift	0.7374	0.6995	0.6653	0.6455	0.7987	0.6450
MARC: Medium Shift	0.7007	0.6866	0.6531	0.6211	0.7791	0.6354
MARC: High Shift	0.6686	0.6784	0.6530	0.6091	0.7659	0.6263
MARC: Extreme Shift	0.6075	0.6364	0.6393	0.5852	0.7102	0.6044
DS: Low Shift	0.7483	0.7019	0.6993	0.6586	0.8294	0.6447
DS: Medium Shift	0.7478	0.6866	0.6924	0.6386	0.8360	0.6404
DS: High Shift	0.7430	0.6458	0.6795	0.6443	0.8338	0.6434
DS: Extreme Shift	0.7482	0.5421	0.6230	0.6436	0.7632	0.6446

These results also offer some interesting conclusions to ponder:

- First and foremost, SCRE is a clear winner, significantly outperforming all the other methods by a wide margin, for almost all types and degrees of shift studied, with the only exception being extreme degrees of MAR bias. It is also the only method that also improves classifier performance when used on non-shifted datasets.
- All the other methods tested perform similarly, with quite poor results: up until high degrees of dataset shift, they perform worse than not using any solver, and only in the most extreme cases is there an improvement by using a dataset shift solver, but even in those cases the difference is minimal.

VI. Concluding remarks

We have proposed a systematic methodology for the generation of artificially shifted datasets, which solves one of the main problems in the field and permits a fair comparison of solutions, which is paramount for the advancement of the field.

We have shown that most classical classifiers can absorb a low degree of dataset shift without much loss in performance, but that once the degree of shift increases classifier performance is significantly affected.

We have also compared the performance of current proposals to work under dataset shift conditions, finding that SCRE performs clearly better than the other studied methods under almost all the conditions tested, and should be considered from now on the state of the art to measure new proposals against. The remaining studied methods perform quite poorly, which leads us to think there is still a large amount of improvement to be made in the field.

The one problem that seems most urgent is the extreme degree MAR bias, where not even SCRE is capable of improving the base classifier performance.

For Peer Review

Acknowledgments

This research was partially funded by projects TIN2011-28488 and P10-TIC-6858 from the MEC of the Spanish government. This work was also supported in part by the National Science Foundation(NSF) Grant ECCS-0926170. Lastly, the work was also partially supported by the Spanish projects DPI2009-08424 and TEC2008-01348/TEC. The authors would like to thank Drs. Bickel, Gretton, Globerson, Shimodaira and Sugiyama for their disposition to share their code.

References

[1] J. Sáez, J. Luengo, and F. Herrera, "A first study on the noise impact in classes for fuzzy rule based classification systems," in *Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference on*. IEEE, 2010, pp. 153–158.

[2] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[3] J. Quiñonero Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[4] J. G. Moreno-Torres, T. Raeder, R. Aláiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognition*, vol. 45, no. 1, pp. 521 – 530, 2012.

[5] J. A. Sáez, J. Luengo, and F. Herrera, "Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification," *Pattern Recognition*, vol. 46, no. 1, pp. 355–364, 2013.

[6] T. Fawcett and P. A. Flach, "A response to Webb and Ting's 'On the application of ROC analysis to predict classification performance under varying class distributions'," *Machine Learning*, vol. 58, no. 1, pp. 33–38, 2005.

[7] R. Alaiz-Rodríguez and N. Japkowicz, "Assessing the impact of changing environments on classifier performance," in *Canadian AI'08: Proceedings of the Canadian Society for computational studies of intelligence, 21st conference on Advances in artificial intelligence*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 13–24.

[8] P. Laskov and R. Lippmann, "Machine learning in adversarial environments," *Machine Learning*, vol. 81, pp. 115–119, 2010.

[9] J. G. Moreno-Torres and F. Herrera, "A preliminary study on overlapping and data fracture in imbalanced domains by means of genetic programming-based feature extraction." in *ISDA*. IEEE, 2010, pp. 501–506.

[10] J. G. Moreno-Torres, J. A. Sáez, and F. Herrera, "Study on the impact of partition-induced dataset shift on k-fold cross-validation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1304–1312, 2012.

[11] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework," *Journal of Multiple-Valued Logic and Soft Computing, In Press*, 2010.

[12] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, pp. 21–27, 1967.

[13] J. R. Quinlan, *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[14] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural Comput.*, vol. 13, no. 3, pp. 637–649, Mar. 2001.

[15] H. Shimodaira, "Improving predictive inference under Covariate Shift by Weighting the Log-likelihood Function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, 2000. [Online]. Available: http://dx.doi.org/10.1016/S0378-3758(00)00115-4

[16] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate Shift Adaptation by Importance Weighted Cross Validation," *Journal of Machine Learning Research*, vol. 8, pp. 985–1005, 2007.

[17] T. Kanamori, S. Hido, and M. Sugiyama, "A least-squares Approach to Direct Importance Estimation," *Journal of Machine Learning Research*, vol. 10, pp. 1391–1445, December 2009.

[18] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning under covariate shift," *Journal of Machine Learning Research*, vol. 10, pp. 2137–2155, September 2009. [Online]. Available: http://www.jmlr.org/papers/volume10/bickel09a/bickel09a.pdf

[19] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf, "Covariate Shift by Kernel Mean Matching," in *Dataset Shift in Machine Learning*, J. Quiñonero Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, Eds. The MIT Press, 2009, pp. 131–160.

[20] R. Alaíz-Rodríguez, A. Guerrero-Curieses, and J. Cid-Sueiro, "Class and subclass probability re-estimation to adapt a classifier in the presence of concept drift," *Neurocomputing*, vol. 74, no. 16, pp. 2614–2623, 2011.

[21] J. G. Moreno-Torres, X. Llorà, D. E. Goldberg, and R. Bhargava, "Repairing Fractures between Data using Genetic Programming-based Feature Extraction: A Case Study in Cancer Diagnosis," *Information Sciences, In Press*, 2010.

Jose G. Moreno-Torres Jose G. Moreno-Torres received the M.Sc. degree in Computer Science in 2008 from the University of Granada, Spain. After spending a year as a fellow of an international "la Caixa" scholarship, during which he did research at the IlliGAL laboratory under the supervision of Prof. David E. Goldberg, he is currently a PhD candidate under the supervision of Prof. Francisco Herrera, working with the Soft Computing and Intelligent Information Systems group in the Department of Computer Science and Artificial Intelligence at the University of Granada. His current research interests include dataset shift, imbalanced classification, and bibliometrics.

Troy Raeder Troy Raeder is a PhD student in Computer Science at the University of Notre Dame in South Bend, IN, USA. His research interests include scenario analysis in machine learning, evaluation methodologies in machine learning, and robust models for changing data distributions. He received B.S. and M.S. degrees in computer science from Notre Dame in 2005 and 2009 respectively.

Rocío Aláiz-Rodríguez Rocío Aláiz-Rodríguez received the BS degree in Electrical Engineering from the University of Valladolid, Spain, in 1999 and the Ph.D. degree from Carlos III University of Madrid, Spain. She is currently an Associate Professor at the Department of Electrical and Systems Engineering, University of Leon, Spain. Her research interests include learning theory, statistical pattern recognition, neural networks and their applications to image processing and quality assessment (in particular, food and frozen-thawed animal semen).

Nitesh V. Chawla Dr. Nitesh Chawla is an Associate Professor in the Department of Computer Science and Engineering at the University of Notre Dame. He directs the Data Inference Analysis and Learning Lab (DIAL) and is also the co-director of the Interdisciplinary Center of the Network Science and Applications (iCenSA) at Notre Dame. His research is supported with research grants from organizations such as the National Science Foundation, the National Institute of Justice, the Army Research Labs, and Industry Sponsors. His research group has received numerous honors, including best papers, outstanding dissertation, and a variety of fellowships. He has also been noted for his teaching accomplishments, receiving the National Academy of Engineers CASEE New Faculty Fellowship, and the Outstanding Undergraduate Teacher Award in 2008 and 2011. He is an Associate Editor for IEEE Transactions of Systems, Man and Cybernetics Part B and Pattern Recognition Letters. More information is available at http://www.nd.edu/ nchawla.

Francisco Herrera Francisco Herrera received his M.Sc. in Mathematics in 1988 and Ph.D. in Mathematics in 1991, both from the University of Granada, Spain.

He is currently a Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada. He has published more than 230 papers in international journals. He is coauthor of the book "Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases" (World Scientific, 2001).

He currently acts as Editor in Chief of the international journal "Progress in Artificial Intelligence" (Springer). He acts as area editor of the International Journal of Computational Intelligence Systems and associated editor of the journals: IEEE Transactions on Fuzzy Systems, Information Sciences, Knowledge and Information Systems, Advances in Fuzzy Systems, and International Journal of Applied Metaheuristics Computing; and he serves as member of several journal editorial boards, among others: Fuzzy Sets and Systems, Applied Intelligence, Information Fusion, Evolutionary Intelligence, International Journal of Hybrid Intelligent Systems, Memetic Computation, and Swarm and Evolutionary Computation.

He received the following honors and awards: ECCAI Fellow 2009, 2010 Spanish National Award on Computer Science ARITMEL to the "Spanish Engineer on Computer Science", International Cajastur "Mamdani" Prize for Soft Computing (Fourth Edition, 2010), IEEE Transactions on Fuzzy System Outstanding 2008 Paper Award (bestowed in 2011), and 2011 Lotfi A. Zadeh Prize Best paper Award of the International Fuzzy Systems Association.

His current research interests include computing with words and decision making, bibliometrics, data mining, data preparation, instance selection, fuzzy rule based systems, genetic fuzzy systems, knowledge extraction based on evolutionary algorithms, memetic algorithms and genetic algorithms.

For Peer Review

2. A proposal to solve Dataset Shift by means of Genetic Programming based Feature Extraction

The journal paper associated to this part is:

2.1. Repairing fractures between data using Genetic Programming-based feature extraction: A case study in cancer diagnosis

- J.G. Moreno-Torres, X. Llorà, D.E. Goldberg, R. Bhargava, Repairing fractures between data using Genetic Programming-based feature extraction: A case study in cancer diagnosis. Information Sciences, 222 (2013) 805-823. doi: 10.1016/j.ins.2010.09.018.

  - Status: **Published**.
  - Impact Factor (JCR 2011): 2.833.
  - Subject Category: Computer Science, Information Systems. Ranking 9 / 135 (Q1).

Information Sciences 222 (2013) 805–823

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Repairing fractures between data using genetic programming-based feature extraction: A case study in cancer diagnosis

Jose G. Moreno-Torres [a,*], Xavier Llorà [b], David E. Goldberg [c], Rohit Bhargava [d]

[a] Department of Computer Science and Artificial Intelligence, Universidad de Granada, 18071 Granada, Spain
[b] National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana-Champaign 1205 W. Clark Street, Urbana, Illinois, USA
[c] Illinois Genetic Algorithms Laboratory (IlliGAL) University of Illinois at Urbana-Champaign 104 S. Mathews Ave, Urbana, Illinois, USA
[d] Department of Bioengineering, University of Illinois at Urbana-Champaign 405 N. Mathews Ave, Urbana, Illinois, USA

ARTICLE INFO

Article history:
Available online 6 October 2010

Keywords:
Genetic programming
Feature extraction
Fractures between data
Biological data
Cancer diagnosis
Different laboratories

ABSTRACT

There is an underlying assumption on most model building processes: given a learned classifier, it should be usable to explain unseen data from the same given problem. Despite this seemingly reasonable assumption, when dealing with biological data it tends to fail; where classifiers built out of data generated using the same protocols in two different laboratories can lead to two different, non-interchangeable, classifiers. There are usually too many uncontrollable variables in the process of generating data in the lab and biological variations, and small differences can lead to very different data distributions, with a fracture between data.

This paper presents a genetics-based machine learning approach that performs feature extraction on data from a lab to help increase the classification performance of an existing classifier that was built using the data from a different laboratory which uses the same protocols, while learning about the shape of the fractures between data that motivated the bad behavior.

The experimental analysis over benchmark problems together with a real-world problem on prostate cancer diagnosis show the good behavior of the proposed algorithm.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

The assumption that a properly trained classifier will be able to predict the behavior of unseen data from the same problem is at the core of any automatic classification process. However, this hypothesis tends to prove unreliable when dealing with biological (or other experimental sciences) data, especially when such data is provided by more than one laboratory, even if they are following the same protocols to obtain it.

The specific problem this paper attempts to solve is the following: we have data from one laboratory (dataset A), and derive a classifier from it that can predict its category accurately. We are then presented with data from a second laboratory (dataset B). This second dataset is not accurately predicted by the classifier we had previously built due to a fracture between the data of both laboratories. We intend to find a transformation of dataset B (dataset S) where the classifier works.

Evolutionary computing, as introduced by Holland [29]; is based on the idea of the survival of the fittest, evoked by the natural evolutionary process. In genetic algorithms (GAs) [22], solutions (genes) are more likely to reproduce the fitter

* Corresponding author. Tel.: +34588998183.
E-mail addresses: jose.garcia.mt@decsai.ugr.es (J.G. Moreno-Torres), xllora@illinois.edu (X. Llorà), deg@illinois.edu (D.E. Goldberg), rbx@uiuc.edu (R. Bhargava).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.09.018

they are, and random sporadic mutations help maintain population diversity. Genetic Programming (GP) [35] is a development of those techniques, and follows a similar pattern to evolve tree-shaped solutions using variable-length chromosomes.

Feature extraction, as defined by Wyse et al. [58], 'consists of the extraction a set of new features from the original features through some functional mapping'. Our approach to the problem can be seen as feature extraction, since we build a new set of features which are functions of the old ones. However, we have a different goal than that of classical feature extraction, since our intention is to fit a dataset to an already existing classifier, not to improve the performance of a future one.

In this work, we intend to demonstrate the use of GP-based feature extraction to unveil transformations in order to improve the accuracy of a previously built classifier, by performing feature extraction on a dataset where said classifier should, in principle, work; but where it does not perform accurately enough. We test our algorithm first on artificially-built problems (where we apply ad hoc transformations to datasets from which a classifier has been built, and use the dataset resulting from those transformations as our problem dataset); and then on a real-world application where biological data from two different medical laboratories regarding prostate cancer diagnosis are used as datasets A and B.

Even though the method proposed in this paper does not attempt to reduce the number of features or instances in the dataset, it can still be regarded as a form of data reduction because it unifies the data distributions of two datasets; which results in the capability of applying the same classifier to both of them, instead of needing two different classifiers, one for each dataset.

The remainder of this paper is organized as follows: in Section 2, some preliminaries about the techniques used and some approaches to similar problems in the literature are presented. Section 3 details the real-world biological problem that motivates this paper. Section 4 has a description of the proposed algorithm GP-RFD; and Section 5 includes the experimental setup, along with the results obtained, and an analysis. Finally, in Section 6 some concluding remarks are made.

2. Preliminaries

This section is divided in the following way: in Subsection 2.1 we introduce the notation that has been used in this paper. Then we include an introduction to GP in Subsection 2.2, a brief summary of what has been done in feature extraction in Subsection 2.3, and a short review of the different approaches we found in the specialized literature on the use of GP for feature extraction in Subsection 2.4. We conclude mentioning some works related to the finding and repair of fractures between data in Subsection 2.5.

2.1. Notation

A classification problem is considered with:

- A set of input variables $X = \{x_i / i = 1, \ldots, n_v\}$, where n_v is the number of features (attributes) of the problem.
- A set of values for the target variable (class) $C = \{C^j / j = \{1, \ldots, n_c\}\}$, where n_c is the number of different values for the class variable.
- A set of examples $E = \{e^h = (e^h_1, \ldots, e^h_{n_v}, C^h) / h = 1, \ldots, n_e\}$, where C^h is the class label for the sample e^h, and n_e is the number of examples.

When describing the problem, we mention datasets A, B and S. They correspond to:

- A: the original dataset that was used to build the classifier.
- B: the problem dataset. The classifier is not accurate on this dataset, and that is what the proposed algorithm attempts to solve.
- S: the solution dataset, result of applying the evolved transformation to the samples in dataset B. The goal is to have the classifier performance be as high as possible on this dataset.

When performing experiments and obtaining the evolved expressions, we use the following notation: when artificially creating a dataset B by means of a fabricated transformation over dataset A, we have $B = \{b_i / i = 1, \ldots, n_v\}$ be the attributes in dataset B and $A = \{a_i / i = 1, \ldots, n_v\}$ be the ones from dataset A. In appendix A, we show the learned transformations for the prostate cancer problem. The attributes shown are those corresponding to dataset S, and are represented as $S = \{s_i / i = 1, \ldots, n_v\}$.

2.2. Genetic programming

A GA [22] is a stochastic optimization technique inspired by nature's development of useful characters. It is based on the idea of survival of the fittest [12] in the following way: given a population of possible solutions to a problem (represented by

chromosomes), there is some selection procedure that favors the fitter ones (i.e., the ones that provide a higher-quality solution); and the selected chromosomes get an opportunity to pass down their genetic material to the next generation via some crossover operator; which usually builds new individuals from the combination of old ones. In some variations of the algorithm, random mutations are sporadically introduced to help maintain biological diversity in the population.

GP, as proposed by John Koza in 1992 [35], uses a selectorecombinative schema where the solutions are represented by trees; which are encoded as variable-length chromosomes. It was originally designed to automatically develop programs, but it has been used for multiple purposes due to its high expressive power and flexibility. In the words of Poli and Langdon [48], 'GP is a systematic, domain-independent method for getting computers to solve problems automatically starting from a high-level statement of what needs to be done. Using ideas from natural evolution, GP starts from an ooze of random computer programs, and progressively refines them through processes of mutation and sexual recombination, until solutions emerge. This is all done without the user having to know or specify the form or structure of solutions in advance. GP has generated a plethora of human-competitive results and applications, including novel scientific discoveries and patentable inventions'.

There are a few details about GP that make it different from standard GAs:

- Crossover: the most commonly used operator is one-point crossover, which is analogous to the GA classical one, but where subtrees instead of a specific gene signal where the cut is made.
- Even though mutation was used in the early literature regarding the evolution of programs (see [7,11,17]) Koza chose not to use it ([35,36]), as he wished to demonstrate that mutation was not necessary. This has significantly influenced the field, and mutation was often omitted from GP runs. However, mutation has proved useful since then (see [5,44], for example); and its use is widely spread nowadays. Multiple different mutation operators have been proposed in the literature [46].
- Treatment of constants: the discovery of constants is one of the hardest issues in GP. Koza proposed a solution called Ephemeral Random Constant (ERC), which uses a fixed terminal (e) to represent a constant. The first time one of such constants is evaluated, it gets assigned a random value. From there on, it retains that value throughout the whole run. A number of alternatives have been proposed in the literature [15,51], but ERC remains the most used one.
- Automatically defined functions: ADFs were also first proposed by Koza [36]. The idea is to permit each individual to evolve more than one tree simultaneously; having the extra trees work as primitives that can be called from the main one.

GP has been applied often to classification [14]. Among the latest advances in the field, we would like to mention those dedicated to high dimensional problems [37,6], variations in population size [33,34], and applications to other related fields [60,3].

2.3. Feature extraction

Feature extraction creates new features as functional mappings of the old ones. It has been used both as a form of preprocessing, which is the approach we use in this paper, and also embedded with a learning process in wrapper techniques. An early proposer of such a term was probably Wyse in 1980, in a paper about intrinsic dimensionality estimation [58]. There are multiple techniques that have been applied to feature extraction throughout the years, ranging from principal component analysis to support vector machines to GAs (see [30,47,45], respectively, for some examples).

Among the foundations papers in the literature, Liu's book in 1998 [40] is one of the earlier compilations of the field. As a result of a workshop held in 2003 [25], Guyon and Elisseeff published a book with an important treatment of the foundations [26].

2.4. Genetic programming-based feature extraction

GP has been used extensively to optimize feature extraction and selection tasks. One of the first contributions in this line was the one published by Tackett in 1993 [55], who applied GP to feature discovery and image discrimination tasks.

We can consider two main branches in the philosophy of GP-based feature extraction:

On one hand, we have the proposals that focus only on the feature extraction procedure, of which there are multiple examples: Sherrah et al. [52] presented in 1997 the evolutionary pre-processor (EPrep), which searches for an optimal feature extractor by minimizing the misclassification error over three randomly selected classifiers. Kotani et al.'s work from 1999 [32] determined the optimal polynomial combinations of raw features to pass to a k-nearest neighbor classifier. In 2001, Bot [8] evolved transformed features, one-at-a-time, again for a k-NN classifier, utilizing each new feature only if it improved the overall classification performance. Zhang and Rockett, in 2006, [63] used multiobjective GP to learn optimal feature extraction in order to fold the high-dimensional pattern vector to a one-dimensional decision space where the classification would be trivial. Lastly, also in 2006, Guo and Nandi [24] optimized a modified Fisher discriminant using GP, and then Zhang et al. extended their work by using a multiobjective approach to prevent tree bloat [64], and applied a similar method to spam filtering [62].

On the other hand, some authors have chosen to evolve a full classifier with an embedded feature extraction step. As an example, Harris [28] proposed in 1997 a co-evolutionary strategy involving the simultaneous evolution of the feature extraction procedure along with a classifier. More recently, Smith and Bull [54] developed a hybrid feature construction and selection method using GP together with a GA. FLGP, by Yin et al. [39] is yet another example, where 'new features extracted by certain layer are used to be the training set of next layer's populations'.

2.5. Finding and repairing fractures between data

Throughout the literature there have been a number proposals to quantify the amount of dataset shift (in other words, the size of the fracture in the data). This shift is usually due to time passing (the data comes from the same source at a latter time), but it can also be due to the data being originated by different sources, as is the case in this paper. Some of the most relevant works in the field are: Wang et al. [56], where the authors present the idea of correspondence tracing. They propose an algorithm for the discovering of changes of classification characteristics, which is based on the comparison between two rule-based classifiers, one built from each dataset. Yang et al. [59] presented in 2008 the idea of conceptual equivalence as a method for contrast mining, which consists of the discovery of discrepancies between datasets. Lately, it is important to mention the work by Cieslak and Chawla [10], which presents a statistical framework to analyze changes in data distribution resulting in fractures between the data.

A different approach to fixing data fractures relies on the adaptation of the classifier. Quiñonero-Candela et al. [49] edited a very interesting book on the topic, including several specific proposals to repair fractures between data (what they call dataset shift). There are two main differences between the usual proposals in the literature and this contribution: first, they are most often based on altering the classifier, while we propose keeping it intact and transforming the data. Second, most authors focus on covariate shift, a specific kind of data fracture, but the method we propose here is more general and can tackle any kind of shift.

3. Case study: prostate cancer diagnosis

This section begins with an introduction to the importance of the problem in Subsection 3.1. The diagnostic procedure is summarized in Subsection 3.2, and the reason to apply GP-RFD to this problem is shown in Subsection 3.3. Finally, the preprocessing the data went through is presented in Subsection 3.4.

3.1. Motivation

Prostate cancer is the most common non-skin malignancy in the western world. The American Cancer Society estimated 192,280 new cases and 27,360 deaths related to prostate cancer in 2009 [2]. Recognizing the public health implications of this disease, men are actively screened through digital rectal examinations and/or serum prostate specific antigen (PSA) level testing. If these screening tests are suspicious, prostate tissue is extracted, or biopsied, from the patient and examined for structural alterations. Due to imperfect screening technologies and repeated examinations, it is estimated that more than one million people undergo biopsies in the US alone.

3.2. Diagnostic procedure

Biopsy, followed by manual examination under a microscope is the primary means to definitively diagnose prostate cancer as well as most internal cancers in the human body. Pathologists are trained to recognize patterns of disease in the architecture of tissue, local structural morphology and alterations in cell size and shape. Specific patterns of specific cell types distinguish cancerous and non-cancerous tissues. Hence, the primary task of the pathologist examining tissue for cancer is to locate foci of the cell of interest and examine them for alterations indicative of disease. A detailed explanation of the procedure is beyond the scope of this paper and can be found elsewhere [38,43,42].

Operator fatigue is well-documented and guidelines limit the workload and rate of examination of samples by a single operator (examination speed and throughput). Importantly, inter- and intra-pathologist variation complicates decision making. For this reason, it would be extremely interesting to have an accurate automatic classifier to help reduce the load on the pathologists. This was partially achieved in [43], but some issues remain open.

3.3. The generalization problem

Llorà et al. [43] successfully applied a genetics-based approach to the development of a classifier that obtained human-competitive results based on FTIR data. However, the classifier built from the data obtained from one laboratory proved remarkably inaccurate when applied to classify data from a different hospital. Since all the experimental procedure was identical; using the same machine, measuring and post-processing; and having the exact same lab protocols, both for tissue extraction and staining; there was no factor that could explain this discrepancy.

What we attempt to do with this work is develop an algorithm that can evolve a transformation over the data from the second laboratory, creating a new dataset where the classifier built from the first lab is as accurate as possible. This evolved transformation would also provide valuable information, since it would allow the scientists processing the tissues analyze the differences between their results and those of other hospitals.

3.4. Pre-processing of the data

The biological data obtained from the laboratories has an enormous size (in the range of 14 GB of storage per sample); and parallel computing was needed to achieve better-than-human results. For this reason, feature selection was performed on the dataset obtained by FTIR. It was done by applying an evaluation of pairwise error and incremental increase in classification accuracy for every class, resulting in a subset of 93 attributes. This reduced dataset provided enough information for classifier performance to be rather satisfactory: a simple C4.5 classifier achieved ∼95% accuracy on the data from the first lab, but only ∼80% on the second one. The dataset consists of 789 samples from one laboratory and 665 from the other one. These samples represent 0.01% of the total data available for each data set, which were selected applying stratified sampling without replacement. A detailed description of the data pre-processing procedure can be found in [16].

4. A proposal for GP-based feature extraction for the repairing of fractures between data (GP-RFD)

This section is presented in the following way: first, a justification for the choice of GP is included. Subsection 4.1 details how the solutions are represented, then the fitness evaluation procedure and the genetic operators are introduced in Subsections 4.2 and 4.3 respectively. Then, the parameter choices are explained in Subsection 4.4, while the function set is in Subsection 4.5. Finally, the execution flow of the whole procedure is shown in Subsection 4.6.

The problem we are attempting to solve is the design of a method that can create a transformation from a dataset (dataset B) where a classification model is not accurate enough into a new one where it is (dataset S). Said classifier is kept unchanged throughout the process.

We decided to use GP to solve the problem for a number of reasons: first, it is well suited to evolve arbitrary expressions because its chromosomes are trees. This is useful in our case because we want to have the maximum possible flexibility in terms of the functional expressions that can be present in the feature extraction procedure. Second, GP provides highly-interpretable solutions. This is an advantage because our goal is not only to have a new dataset where the classifier works, but also to analyze what was the problem in the first dataset.

The specific decisions to be made once GP was chosen as the technique to apply are how to represent the solutions, what terminals and operators to choose, how to calculate the fitness of an individual and which evolutionary parameters (population size, number of generations, selection and mutation rates, etc.) are appropriate for each specific problem. To clarify some of the points, let us have a binary 2-dimensional problem as an example, and let us use a function set composed of $\{+,-,*,\div\}$.

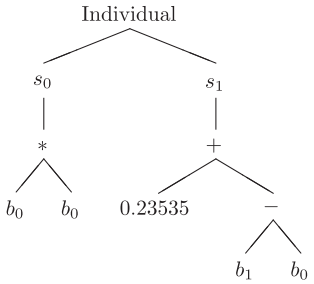
4.1. Solutions representation: context-free grammar

The representation issue was solved by extending GP to evolve more than one tree per solution. Each individual is composed by n trees, where $n = n_v$, the number of attributes present in the dataset (we are trying to develop a new dataset with the same number of attributes as the old one). In the tree structure, the leaves are either constants (we use the Ephemeral Random Constant approach) or attributes from the original dataset. The intermediate nodes are functions from the function set, which is specific to each problem.

The attributes on the transformed dataset are represented by algebraic expressions. These expressions are generated according to the rules of a context-free grammar which allows the absence of some of the functions or terminals. The grammar corresponding to the example problem would look like this:

$$
\begin{aligned}
&\mathit{Start} \rightarrow \mathit{Tree}\ \mathit{Tree}\\
&\mathit{Tree} \rightarrow \mathit{Node}\\
&\mathit{Node} \rightarrow \mathit{Node}\ \mathit{Operator}\ \mathit{Node}\\
&\mathit{Node} \rightarrow \mathit{Terminal}\\
&\mathit{Operator} \rightarrow +|-|*|\div\\
&\mathit{Terminal} \rightarrow x_0|x_1|E\\
&E \rightarrow \mathit{realNumber}(\mathit{represented\ by}\ e)
\end{aligned}
$$

An individual in the example problem would have two trees; and each of them would be allowed to have any of the functions in the function set, which for this example is $\{+,-,*,\div\}$, in their intermediate nodes; and any of $\{x_0, x_1, e\}$ in the leaves. This, for example, would be a legal individual:



4.2. Fitness evaluation

The fitness evaluation procedure is probably the most treated aspect of design in the literature when dealing with GP-based feature extraction. As has been stated before, the idea is to have the provided classifier's performance drive the evolution. To achieve that, GP-RFD calculates fitness in the following way:

1. Prerequisite: a previously built classifier (the one built from dataset A) needs to be provided. It is used as a black box.
2. Given an individual composed of a list of expression trees (one corresponding to each extracted attribute), a new dataset (dataset S) is built applying the transformations encoded in those expression trees to all the samples in dataset B.
3. The fitness of the individual is the classifier's accuracy on dataset S (training-set accuracy), calculated as the ratio of correctly classified samples over the total number of samples.

Fig. 1 presents a schematic representation of the procedure.

4.3. Genetic operators

This section details the choices made for selection, crossover and mutation operators. Since the objective of this work is not to squeeze the maximum possible performance from GP, but rather to show that it is an appropriate technique for the problem and that it can indeed solve it, we did not pay special attention to these choices, and picked the most common ones in the specialized literature.

- Tournament selection without replacement. To perform this selection, k individuals are first randomly picked from the population (where k is the tournament size), while avoiding using any member of the population more than once. The selected individual is then chosen as the one with the best fitness among those picked in the first stage.
- One-point crossover: for each dimension, a subtree from one of the parents is substituted by one from the other parent. The procedure is specified in Algorithm 1. An example, for one of the dimensions only, can be seen in Fig. 2.

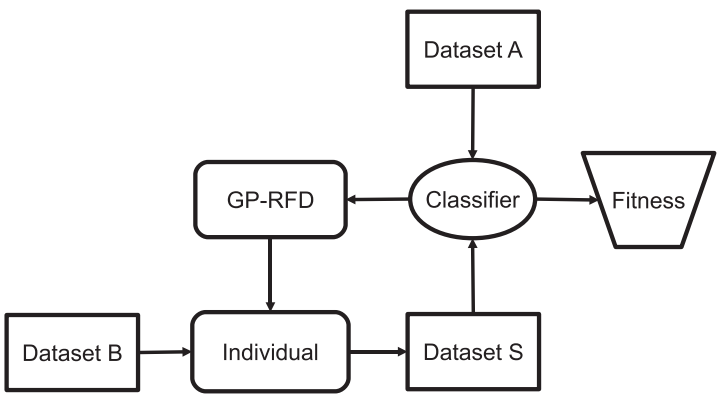


Fig. 1. Schematic representation of the fitness evaluation procedure.

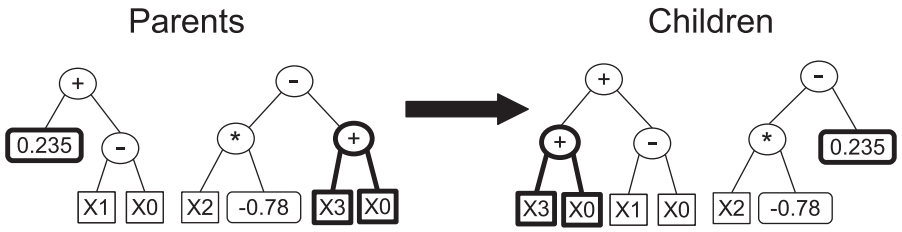


Fig. 2. Crossover example for one of the dimensions only, this is repeated for all dimensions (trees) on each individual.

- Swap mutation: this is a conservative mutation operator, that helps diversify the search within a close neighborhood of a given solution. It consists of exchanging the primitive associated to a node by one that has the same number of arguments.

Algorithm 1. One-point crossover procedure

FORALL trees on each individual
1. Randomly select a non-root non-leave node on each of the two parents.
2. The first child is the result of swapping the subtree below the selected node in the father for that of the mother.
3. The second child is the result of swapping the subtree below the selected node in the mother for that of the father.

- Replacement mutation: this is a more aggressive mutation operator that leads to diversification in a larger neighborhood. The procedure to perform this mutation is the following:
  1. Randomly select a non-root non-leave node on the tree to mutate.
  2. Create a random tree of depth no more than a fixed maximum depth. This parameter has not been tinkered with, since the goal of this study was not to squeeze the maximum performance out of the proposed method, but rather to check its viability. Future work could tackle this issue.
  3. Swap the subtree below the selected node for the randomly generated one.

4.4. Parameters

The evolutionary parameters that were used for the experimental study are detailed in Table 1. As it was mentioned before, not much attention was payed to optimizing the parameters. Because of this the crossover and mutation probabilities, along with the number of generations to run, were fixed to the usual values in the literature (we could call them 'default values') and were not changed in any of the experiments.

Some of the evolutionary parameters are problem dependent, to select an appropriate value for them we used the following rules:

- Population size: since the only measure of difficulty we know about each of our problems a priori is the number of attributes present in the dataset (n_v), we have to fix the population size as a function of it. In the experiments carried out in this study, we found $400{*}n_v$ to be a large enough population to achieve satisfactory results. This parameter is problem-dependent, so what we are fixing here is an upper bound for the population size needed. We found that, by following this

Table 1
Evolutionary parameters for a n_v-dimensional problem.

Parameter	Value
Number of trees	n_v
Population size	$400{*}n_v$
Duration of the run	50 generations
Selection operator	Tournament without replacement
Tournament size	$log_2(n_v) + 1$
Crossover operator	One-point crossover
Crossover probability	0.9
Mutation operator	Replacement & Swap mutations
Replacement mutation probability	0.001
Swap mutation probability	0.01
Maximum depth of the swapped in subtree	5
Function set	Problem dependent
Terminal set	$\{x_0, x_1, \ldots, x_{n_v} - 1, \mathrm{e}\}$

rule, GP-RFD consistently achieved good results; being able to solve the harder transformations, even though it was excessive for the easier ones and thus resulted in slower execution times. If harder problems than the ones studied in this paper were to be tackled, this parameter might need to be revised.

- Tournament size: since we are increasing the population size as a function of n_v, an increase of the selection pressure is needed too. The formula we used to calculate tournament size is: $log_2(n_v) + 1$. Again, this empirical estimation produced the best results; while an excessive pressure produced too fast of a convergence into local optima, and not enough pressure prevents GP-RFD from converging at all.

Table 2
Datasets used.

Dataset	Attributes	Samples	Classes	Class distribution	Attr. type
Linear synthetic	2	1000	2	50–50%	Real
Tao	2	1888	2	50–50%	Real
Iris	4	150	3	33–33–33%	Real
Phoneme	5	5404	2	70–30%	Real
Wisconsin	9	683	2	65–35%	Real
Heart	13	270	2	55–45%	Real
Wine	13	178	3	33–39%–27%	Real
Wdbc	30	569	2	65–45%	Real
Ionosphere	34	351	2	65–45%	Real
Sonar	60	208	2	54–46%	Real
Mux-11	11	2048	2	50–50%	Nominal
Cancer (A)	93	789	2	60–40%	Real
Cancer (B)	93	665	2	60–40%	Real

Table 3
Transformations performed on the Tao dataset.

Experiment	Rotation	Translate & extrude
Transformation applied	$b_0 = a_0 * cos(1) + a_1 * sin(1)$ $b_1 = a_0 * sin(1) + a_1 * cos(1)$	$b_0 = a_0 * 3 + 2$

Table 4
Transformations performed on the UCI and ELENA datasets.

Dataset	In-set transformation	Out-of-set transformation
Iris	$b_2 = a_2 + a_2$	$b_3 = e^{a_3}$
Phoneme	$b_0 = a_0 - 0.4$ $b_3 = a_3 * 2.5$	$b_0 = sin(a_0)$ $b_3 = cos(a_3)$
Wisconsin	$b_1 = a_1 + 2$ $b_5 = a_5 * 3$	$b_1 = cos(a_1)$ $b_5 = sin(a_5)$
Heart	$b_2 = a_2 * 2$ $b_1 1 = a_1 1 + 3$	$b_2 = sin(a_2)$ $b_1 1 = e^{a_1 1}$
Wine	$b_9 = a_9 - 1$ $b_1 2 = a_1 2 * 2$	$b_9 = sin(a_9)$ $b_1 2 = cos(a_1 2)$
Wdbc	$b_2 6 = a_2 6 - 1$ $b_2 7 = a_2 7 * 3$	$b_2 6 = sin(a_2 6)$ $b_2 7 = cos(a_2 7)$
Ionosphere	$b_4 = a_4 - 0.5$ $b_7 = a_7 * 2$	$b_4 = e^{a_4}$ $b_7 = sin(a_7)$
Sonar	$b_7 = a_7 + 0.3$ $b_4 3 = a_4 3 * 2$	$b_7 = sin(a_7)$ $b_4 3 = e^{a_4 3}$

Table 5
Transformations performed on the Multiplexer-11 dataset.

Experiment	Bit flip	Column swap
Transformation applied	$b_1 = not(a_1)$	$b_1 = a_2$ $b_2 = a_3$ $b_3 = a_1$

Table 6
Experimental parameters.

Dataset	Population size	Tournament size	Function set
Linear synthetic	800	2	$\{+,-,*,\div\}$
Tao	800	2	$\{+,-,*,\div\}$
Iris	1600	3	$\{+,-,*,\div\}$
Phoneme	2000	3	$\{+,-,*,\div\}$
Wisconsin	3600	4	$\{+,-,*,\div\}$
Heart	5200	4	$\{+,-,*,\div\}$
Wine	5200	4	$\{+,-,*,\div\}$
Wdbc	12,000	5	$\{+,-,*,\div\}$
Ionosphere	13,600	6	$\{+,-,*,\div\}$
Sonar	24,000	6	$\{+,-,*,\div\}$
Mux-11	4400	4	$\{+,-,*,\div\}$
Cancer	37,200	6	$\{+,-,*,\div,exp,cos\}$

4.5. Function set

Which functions to include in the function set are usually dependent on the problem. , , , Since one of our goals is to have an algorithm as universal and robust as possible, where the user does not need to fine-tune any parameters to achieve good performance; we decided not to study the effect of different function set choices. The used function sets are chosen to be close to the default ones most authors use in the literature, and were extracted in all cases from $\{+,-,*,\div,exp,cos\}$. The benchmark experiments did not use $\{exp,cos\}$, since we intended to test the capability of the method to unveil transformations that did not include functions in the function set. The specific choices for each of the experiments can be seen in Table 6.

4.6. Execution flow

Algorithm 2 contains a summary of the execution flow of the GP procedure, which follows a classical evolutionary scheme. It stops after a user-defined number of generations, providing as a result the best individual (i.e., transformation) it has ever found.

Algorithm 2. Execution flow of the GP procedure

1. Randomly create the initial population by applying the context-free grammar presented in Subsection 4.1.
2. Repeat Ng times (where *Ng* is the number of generations)
   2.1 Evaluate the current population, using the procedure shown in Subsection 4.2.
   2.2 Apply selection and crossover to create a new population that will replace the old one.
   2.3 Apply the mutation operators to the new population.
3. Return the best individual ever seen.

5. Experimental study

This section is organized in the following way: to begin with, a general description of the experimental procedure is presented in Subsection 5.1, along with the datasets that we have used for our testing (both the benchmark problems and the prostate cancer dataset); and also in the benchmarks' case the transformations performed on each of them. The parameters used for each experiment are shown in Subsection 5.2; followed by a presentation of the benchmark experimental results in Subsection 5.3. Finally, the results obtained on the prostate cancer problem are presented in Subsection 5.4.

5.1. Experimental framework, datasets and transformations

The goal of the experiments was to check how effective GP-RFD was in finding a transformation over dataset B that would increase the provided classifier's accuracy. To validate our results, we employed a 5-fold cross validation technique [31]. We used the beagle library [18] for our GP implementation.

The experimental study is fractioned in two parts. In the first one, a synthetic set of tests is built from a few well-known benchmark datasets. The procedure followed in these experiments was (see Fig. 3 for a schematic representation):

1. Split the original dataset in two halves with equal class distribution.
2. Consider the first half, to be dataset A.
3. From dataset A, build a classifier. We chose C4.5 [50], but any other classifier would work exactly the same; due to the fact that GP-RFD uses the learned classifier as a black box.

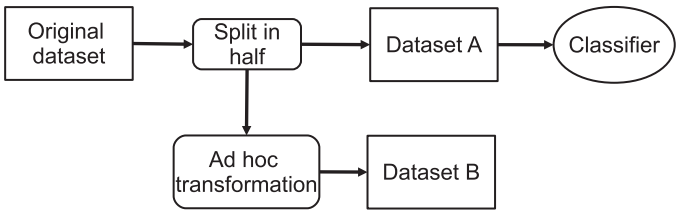


Fig. 3. Schematic representation of the experimental procedure with benchmark datasets.

4. Apply a transformation over the second half of the original dataset, creating dataset B. The transformations we tested were designed to check GP-RFD's performance on different types of problems, including both linear and non-linear transformations. A description of each of them can be found in the next subsection.
5. The performance of the classifier built in step 2 is significantly worse on dataset B than it is on dataset A. This is the starting point on the real problem we are emulating.
6. Apply GP-RFD to dataset B in order to evolve a transformation that will create a solution dataset S. Use 5-fold cross validation over dataset S, so that training and test set accuracy results can be obtained.
7. Check the performance of the step 2 classifier on dataset S. Ideally, it should be close to the one on dataset A, which would mean GP-RFD has successfully discovered the hidden transformation and inverted.

The second part of the study is the application of the proposed algorithm to the prostate cancer problem. The steps followed in this case were:

1. Consider each of the provided datasets to be datasets A and B respectively.
2. From dataset A, build a classifier. Use 5-fold cross validation to obtain training and test-set performance results.
3. Apply GP-RFD to dataset B in order to evolve a transformation that will create a solution dataset S. Use 5-fold cross validation over dataset S, so that training and test set accuracy results can be obtained.
4. Check the performance of the step 2 classifier on dataset S. Ideally, it should be close to the one on dataset A, meaning GP-RFD has successfully discovered the hidden transformation and inverted it.

The selected datasets are summarized in Table 2. A short description and motivation for each of the datasets follows, and this subsection is concluded with the specification of the transformations that were fabricated to test the algorithm on each of the benchmark datasets. For the two-dimensional problems, the transformations are also graphically represented.

Note that the transformations in the prostate cancer problem are not specified. This is due to it being a real-world problem and not a fabricated one, so the implicit transformations in the data were unknown a priori.

- Linear synthetic dataset: we have called the first dataset 'Linear synthetic'. It was created specifically for this work, with the idea of having an easily representable linearly separable dataset to work with. It was chosen to check the performance of GP-RFD on some simple transformations, without the added difficulty of having a complex original dataset. The dataset can be seen in Fig. 4. We applied three transformations to this dataset A: rotation, translation and extrusion and circle. The transformed datasets (datasets B on the experiments) can be seen in Figs. 5–7 respectively.

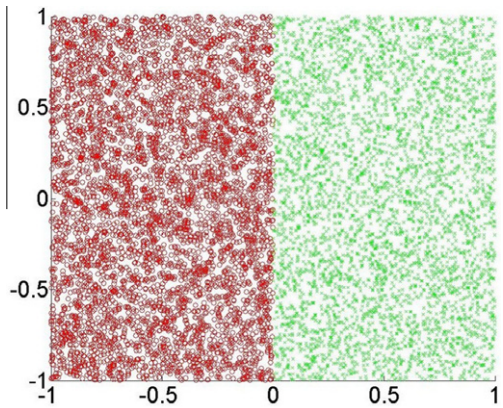


Fig. 4. Linear synthetic dataset, dataset A.

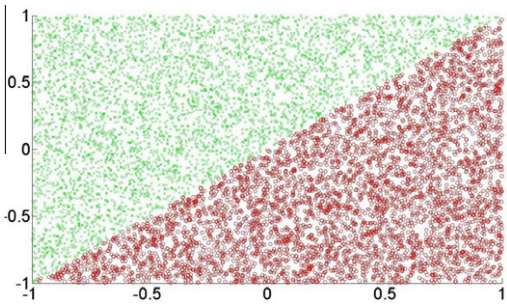


Fig. 5. Rotation problem, transformed dataset.

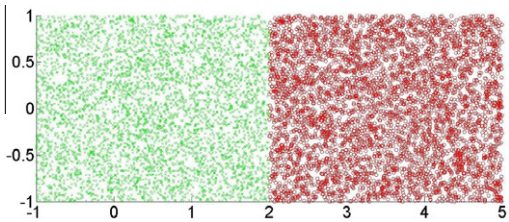


Fig. 6. Translation & extrusion problem, transformed dataset.

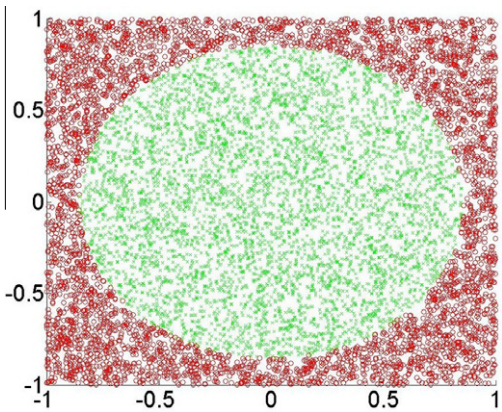


Fig. 7. Circle problem, transformed dataset.

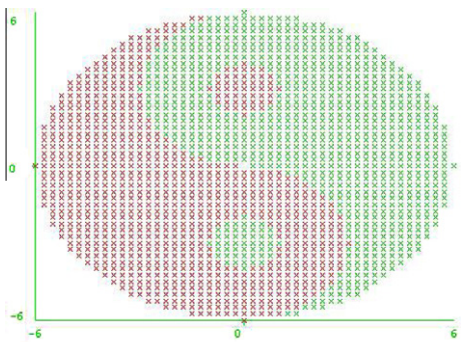


Fig. 8. Tao dataset. This is dataset A, over which the different transformations are applied, and the transformed datasets have to fit to the same classifier this dataset does.

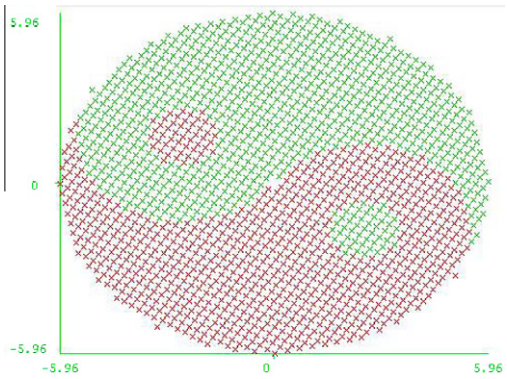


Fig. 9. Rotated Tao, transformed dataset.

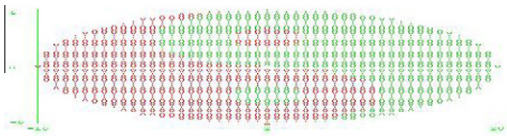


Fig. 10. Translated and extruded Tao, transformed dataset.

- Tao: the next step to check the usefulness of GP-RFD is starting from a harder dataset. To this end, we chose the Tao dataset, still a 2-dimensional problem but where classification is much harder. This dataset is also built artificially [41]. The dataset can be seen, before any transformations (dataset A), in Fig. 8. Mirroring the transformations applied over the linear synthetic dataset, we chose to transform the original Tao dataset by rotating it (Fig. 9); or by translating and extruding (Fig. 10). The transformations applied to Tao can also be seen in Table 3.
- UCI and ELENA datasets: once GP-RFD has been tested in small (with a low number of attributes) datasets, it is useful to see how it fares in bigger benchmark problems. We chose a few different datasets from the UCI database [4], as well as the ELENA project [23]:
  - Iris: classification of iris plants (UCI).
  - Phoneme: distinguish between nasal and oral sounds (ELENA).
  - Wisconsin: diagnosis of breast cancer patients (UCI).
  - Heart: detect the absence or presence of heart disease (UCI).
  - Wine: classification of different types of Italian wines (UCI).
  - Wdbc: determination of whether a found tumor is benign or malignant (UCI).
  - Ionosphere: radar data where the task is to decide is a given radar return is good or bad (UCI, modified as found in the KEEL database [1]).
  - Sonar: distinguishing between rocks and metal cylinders from sonar data (UCI).

We performed two different experiments on each of the datasets. In the first experiment, the transformation is created using functions that appear in the function set of the GP procedure (more specifically, one of the attributes is added to itself). We named this experiment 'in-set transformation'. The second one transforms the dataset by using functions that do not appear in the GP function set. The name for this experiment is 'out-of-set transformation'. The exact details for these transformations can be found in Table 4. Any attribute not specified as being part of the transformation in the tables is assumed to be unchanged.

- Multiplexer-11: since GP-RFD should be flexible enough to be able to tackle datasets with nominal attributes, one of these datasets was included in the testing. In this work, we chose the Multiplexer problem. This is a binary problem where some of the bits act as address, and the remaining bits are data registers. The correct classification for a given input is the value of the register pointed by the address bits. The specific instance used here is Multiplexer-11, a dataset with 11 binary attributes (where the first three act as address, and the remaining eight as registers); and $2^{11} = 2048$ samples. Two different transformations were tested: in the first one, of the address bits was flipped; while in the second experiment there was an attribute swap, in a circular shift. The details can be found in Table 5.
- Prostate cancer: as was explained in Section 3, the solution to this problem is the main motivation for this work. Since we were provided with data from two real laboratories, there was no need to fabricate any transformations: we chose one the data from one of the laboratories as dataset A and the other one as dataset B.

5.2. Parameters

In this section, we detail the parameters used for each of the datasets, including both the evolutionary parameters and the GP setup. The parameters were chosen following the rules detailed in Section 4.4.

As can be seen in Table 6, the population sizes are large. This is mostly due to GP being a technique that traditionally requires large population sizes to be effective, a factor which is aggravated by the fact that GP-RFD evolves multiple expression trees simultaneously (one for each attribute in the dataset). We acknowledge this issue provokes long execution times for some of the experiments, but considered it a secondary concern and did not address it in this work.

5.3. Experimental results: benchmark problems

This part presents the results obtained in terms of classifier performance for the benchmark problems, along with a statistical analysis to evaluate whether GP-RFD is effective.

Table 7 details the performance obtained by the C4.5 classifier on each of the benchmark problems. It includes the classifier performance, calculated as shown on Subsection 4.2, on:

- Dataset A, which was used to generate the decision tree. A 5-fold cross validation technique was applied, and both training and test set results are presented.
- Dataset B, which was created by designing an ad hoc transformation.
- Dataset S, which is the result of applying GP-RFD to dataset B, obtaining a transformed dataset where classifier performance is increased. A 5-fold cross validation technique was applied, and both training and test set results are presented.

The results show that GP-RFD is capable of reversing nearly all of the fabricated transformations, achieving accuracy rates that are very close to the ones obtained in the original datasets in both training and test performances. GP-RFD has also proven capable of generalizing well, as can be seen by the small difference between training and test set classification performances in most cases. However, some of the datasets (which, coincidentally, tend to also behave badly in terms of generalization when building classifiers) present some generalization issues, leading to the inability to fully solve the problem dataset.

5.3.1. Statistical analysis

To complete the experimental study, we have performed a statistical comparison between the classifier performance over the following datasets:

- Dataset A, from which the classifier was built.
- Dataset B, artificially built by injecting an ad hoc transformation.

Table 7
Classifier performance results: benchmark problems.

Problem	Classifier performance on dataset ...				
	A-training	A-test	B	S-training	S-test
Linear synthetic – rotation	1.00000	1.00000	0.24930	1.00000	1.00000
Linear synthetic – translation& extrusion	1.00000	1.00000	0.34160	1.00000	0.99800
Linear synthetic – circle	1.00000	1.00000	0.49860	0.96050	0.94400
Tao – rotation	0.98518	0.93750	0.62924	0.94418	0.94255
Tao – translation& extrusion	0.98518	0.93750	0.80403	0.95344	0.93192
Iris – in-set functions	0.97330	0.93333	0.66667	0.99333	0.92000
Iris – out-of-set functions	0.97330	0.93333	0.60000	0.99000	0.92000
Phoneme – in-set functions	0.91895	0.84160	0.75204	0.828978	0.769907
Phoneme – out-of-set functions	0.91895	0.84160	0.59141	0.839871	0.804815
Wisconsin – in-set functions	0.97361	0.93842	0.35380	0.98248	0.93821
Wisconsin – out-of-set functions	0.97361	0.93842	0.88889	0.98321	0.94412
Heart – in-set functions	0.89630	0.72593	0.45296	0.92778	0.79259
Heart – out-of-set functions	0.89630	0.72593	0.60000	0.96296	0.72594
Wine – in-set functions	0.97727	0.89733	0.65556	0.98889	0.90000
Wine – out-of-set functions	0.97727	0.89733	0.40000	0.96944	0.91111
Wdbc – in-set functions	0.98571	0.92143	0.57143	0.98839	0.946428
Wdbc – out-of-set functions	0.98571	0.92143	0.82857	0.98214	0.97500
Ionosphere – in-set functions	0.98286	0.87429	0.70857	0.98571	0.88571
Ionosphere – out-of-set functions	0.98286	0.87429	0.77714	0.98571	0.857143
Sonar – in-set functions	0.93939	0.60601	0.61000	0.95500	0.66000
Sonar – out-of-set functions	0.93939	0.60601	0.51000	0.94750	0.72000
Mux11 – bit flip	1.00000	0.97070	0.50000	0.96951	0.96667
Mux11 – column swap	1.00000	0.97070	0.62500	0.97195	0.96765

- Dataset S-test, the result of applying GP-RFD over dataset B (test-set results).

In [13,21,19,20] a set of simple, safe and robust non-parametric tests for statistical comparisons of classifiers are recommended. One of them is the Wilcoxon signed-ranks test [57,53], which is the test that we have selected to do the comparison.

This is analogous to the paired t-test in non-parametric statistical procedures; therefore it is a pairwise test that aims to detect significant differences between two sample means, that is, the behavior of two algorithms. It is defined as follows: let d_i be the difference between the performance scores of the two classifiers on the *ith* dataset out of N_{ds} datasets. The differences are ranked according to their absolute values; average ranks are assigned in the case of ties. Let R^+ be the sum of ranks for the data-sets in which the first algorithm outperformed the second, and R^- the sum of ranks for the opposite. Ranks of $d_i = 0$ are split evenly among the sums; if there is an odd number of them, one is ignored:

$$R^+ = \sum_{d_i>0} ank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i)$$
$$R^- = \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \quad (1)$$

Let T be the smaller of the sums, $T = min(R^+, R^-)$. If T is less than or equal to the value of the distribution of Wilcoxon for N_{ds} degrees of freedom [61], the null hypothesis of equality of means is rejected; this will mean that a given classifier outperforms their opposite, with the p-value associated.

The Wilcoxon signed-ranks test is more sensitive than the t-test. It assumes commensurability of differences, but only qualitatively: greater differences still count for more, which is probably desired, but the absolute magnitudes are ignored. From a statistical point of view, the test is safer since it does not assume normal distributions. Also, outliers (extremely good/bad performances) have a smaller effect on the Wilcoxon signed-ranks test than on the t-test.

When the assumptions of the paired t-test are met, the Wilcoxon signed-ranks test is less powerful than the paired t-test. On the other hand, when the assumptions are not met, the Wilcoxon test is a better choice than the t-test. This is because the Wilcoxon test can be applied over the averaged results obtained by the algorithms in each data set, without any assumptions about the characteristics of the distribution of the results obtained.

A complete description of the Wilcoxon signed ranks test and other non-parametric tests for pairwise and multiple comparisons, together with software for their use, can be found in the website available at http://sci2s.ugr.es/sicidm/.

As it was mentioned above, the test was applied to compare the classifier performance in datasets A, B and S. The results can be seen in Table 8. Note that we compare the results in dataset A against those in S both in terms of training and test sets. However, since the classifier was not built from dataset B, we consider those results test-set related and compare it with S-test.

So we can conclude GP-RFD is capable of finding transformations resulting in a new dataset S that

1. Significantly outperforms dataset B in terms of classifier performance.
2. Obtains statistically equivalent results to dataset A, both in terms of training and test sets. Since the classifier was built from dataset A, this means dataset S is a successful repair of the fracture between datasets A and B, assuming class dis-

Table 8
Wilcoxon signed-ranks test results: Benchmark problems.

Comparison	R^+	R^-	p-Value	Null hypothesis of equality
A-test vs B	275	1	4.77E−007	*rejected* (A-test outperforms B)
B vs S-test	0	276	2.38E−007	*rejected* (S-test outperforms B)
A-training vs S-training	147.5	128.5	–	*accepted*
A-test vs S-test	128.5	147.5	–	*accepted*

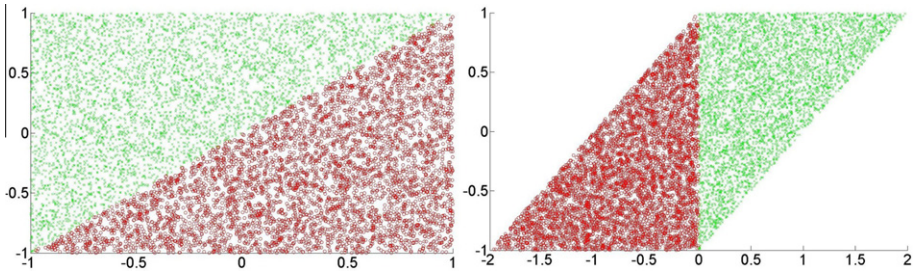


Fig. 11. Linear synthetic rotation, problem (L) and solution (R) datasets.

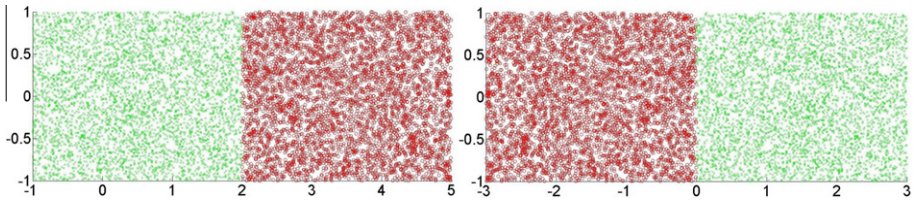


Fig. 12. Linear synthetic translation and extrusion, problem (L) and solution (R) datasets.

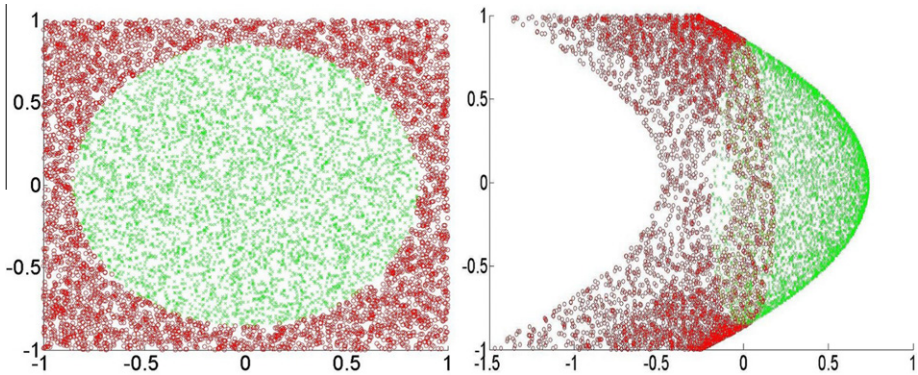


Fig. 13. Circle, problem (L) and solution (R) datasets.

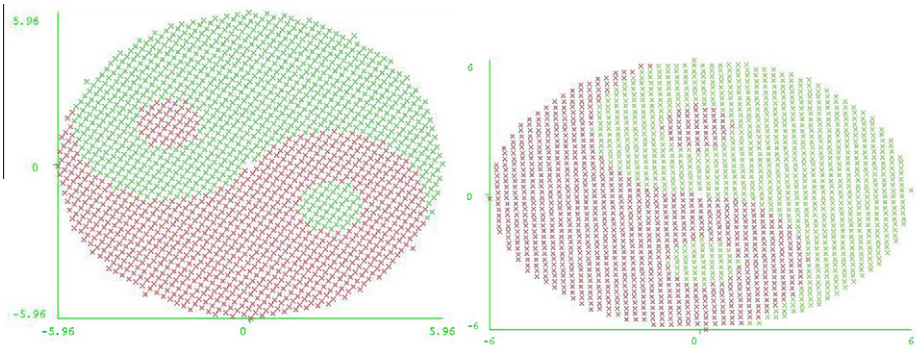


Fig. 14. Rotation in Tao, problem (L) and solution (R) datasets.

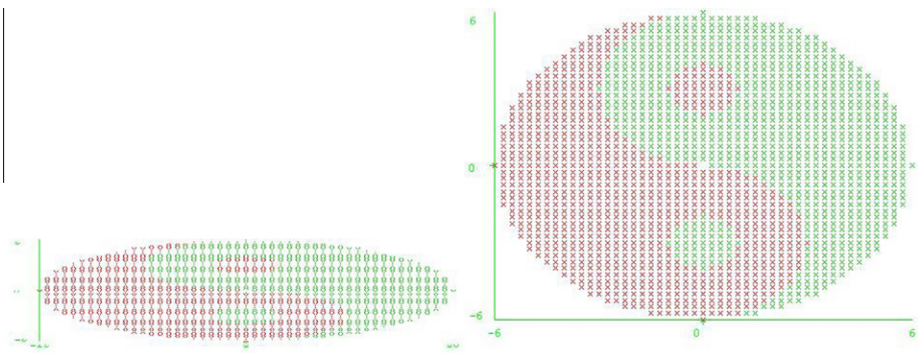


Fig. 15. Translation and extrusion in Tao, problem (L) and solution (R) datasets.

tribution did not change. We know this is the case in these experiments due to the way we built datasets A and B, but it has to be kept in mind when applying the method in other environments.

5.3.2. Graphical results

This section presents graphical representations of some of the obtained results. Since several of the datasets have a high number of variables that make them extremely hard to chart in a simple way, only the results corresponding to the linear synthetic dataset (Figs. 11–13) and the Tao dataset (Figs. 14 and 15) are shown. To make the visualization easier, each of the solution datasets (datasets S) is presented side-by-side with the corresponding problem dataset (datasets B). The original datasets (datasets A) can be seen in Fig. 4 for the linear synthetic dataset and Fig. 8 for the Tao dataset.

5.4. Prostate cancer experimental results

This section presents the preliminary results for the Prostate Cancer problem, in terms of classifier accuracy. The results obtained can be seen in Table 9. In that table, dataset A is the one from the first lab; which was used to build the classifier, dataset B is the one coming from the second lab, and dataset S is the result of the application of GP-RFD.

To check whether the full dataset B was needed to evolve an effective transformation, we also tested using just half of it to train GP-RFD, and the other half to test (2-fold cross validation). These results are also included in Table 9.

The performance results are excellent for a number of reasons. First and foremost, GP-RFD was able to find a transformation over the data from the second laboratory that made the classifier work just as well as it did on the data from the first lab, effectively finding the hidden perturbations that prevented the classifier from working accurately.

The second positive conclusion to be obtained from the results is the generalization power of GP-RFD. As can be observed from the test results, GP-RFD does not 'cheat' by over-learning on the known data, and works well when transforming new, previously unseen, samples.

Third, the results show GP-RFD was capable of obtaining excellent results using just half of the B dataset to train. This result highlights the power of the method to unveil the hidden transformation from a relatively low number of samples.

We also performed a Wilcoxon signed-ranks test to evaluate the performance of GP-RFD over the case of study problem. In order to do it, we used the results from each partition in the 5-fold cross validation procedure. We ran the experiment four times, resulting in $4 * 5 = 20$ performance samples to carry out the statistical test. As we did before, R^+ corresponds to the first algorithm in the comparison winning, and R^- to the second one. Table 10 shows the results.

The results on the case study problem are exactly the same as those achieved in the benchmark problems. We can then conclude GP-RFD was capable of repairing the existing fracture between the data from both laboratories. Again, this conclusion assumes class distribution did not change. It is a given in this case, since we know the class distribution to be equal in datasets A and B, but is an issue that has to be kept in mind when applying the method to other problems.

6. Concluding remarks

We have presented GP-RFD, a new algorithm that approaches a common problem in real life for which not many solutions have been proposed in evolutionary computing. The problem in question is the repairing of fractures between data by adjusting the data itself, not the classifiers built from it.

We have developed a solution to the problem by means of a GP-based algorithm that performs feature extraction on the problem dataset driven by the accuracy of the previously built classifier.

Table 9
Classifier performance results: the prostate cancer problem.

Validation method	Classifier performance in dataset ...				
	A-training	A-test	B	S-training	S-test
5-fold cross validation	0.95435	0.92015	0.83570	0.95191	0.92866
2-fold cross validation	0.95435	0.92015	0.83570	0.95482	0.93223

Table 10
Wilcoxon signed-ranks test results: the prostate cancer problem.

Comparison	R^+	R^-	p-Value	Null hypothesis of equality
A-test vs B	210	0	1.91E−007	*rejected* (A-test outperforms B)
B vs S-test	0	210	1.91E−007	*rejected* (S-test outperforms B)
A-training vs S-training	126	84	–	*accepted*
A-test vs S-test	84	126	–	*accepted*

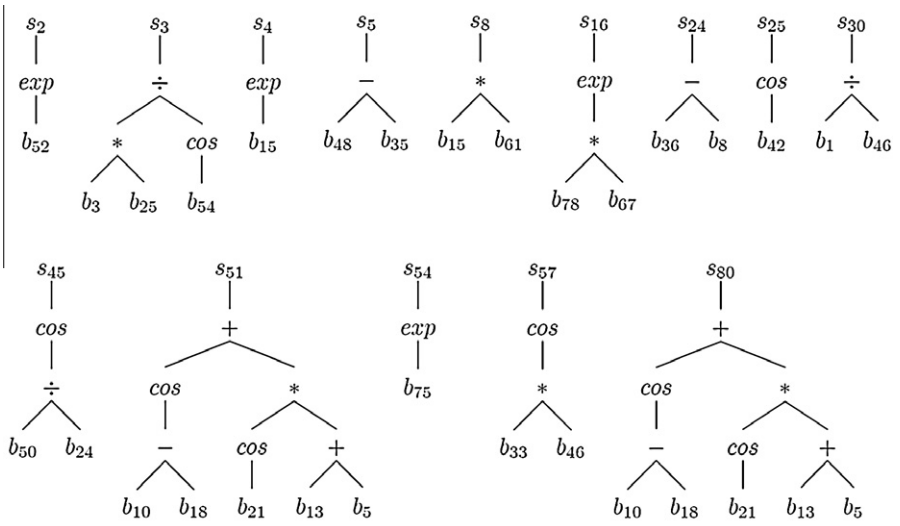


Fig. 16. Tree representation of the expressions contained in a solution to the prostate cancer problem.

We have tested GP-RFD on a set of artificial benchmark problems, where a problem dataset is fabricated by applying an ad hoc disruption to an original dataset, and it has proved capable of solving all the transformations presented showing good performance both in train and, more importantly, test data.

We have also being able to apply GP-RFD to a real-world problem where data from two different laboratories regarding prostate cancer diagnosis was provided, and where the classifier learned from one did not perform well enough on the other. Our algorithm was capable of learning a transformation over the second dataset that made the classifier fit just as well as it did on the first one. The validation results with 5-fold cross validation also support the idea that the algorithm is obtaining good results; and has a strong generalization power.

Lastly, we have applied a statistical analysis methodology that supports the claim that the classifier performance obtained on the solution dataset significantly outperforms the one obtained on the problem dataset.

There is, however, one point where the proposed method has not been successful. The learned transformations have failed to provide any information about why the fracture appeared between the data from the two laboratories. We have, however, included a sample of the transformations learned in appendix A.

Acknowledgments

Jose García Moreno-Torres was supported by a scholarship from 'Obra Social la Caixa' and is currently supported by a FPU grant from the Ministerio de Educación y Ciencia of the Spanish Government, and also by the KEEL project (TIN2008-06681-C06-01). Rohit Bhargava would like to acknowledge collaborators over the years, especially Dr. Stephen M. Hewitt and Dr. Ira W. Levin of the National Institutes of Health, for numerous useful discussions and guidance. Funding for this work was provided in part by University of Illinois Research Board and by the Department of Defense Prostate Cancer Research Program. This work was also funded in part by the National Center for Supercomputing Applications and the University of Illinois, under the auspices of the NCSA/UIUC faculty fellows program.

Appendix A. Sample solution from the prostate cancer problem

In this appendix, we include a sample of the learned transformations for the prostate cancer problem, presenting the transformations corresponding to the highest fitness individual ever found. Due to space concerns, only the attributes relevant to the C4.5 classifier are shown (Fig. 16).

References

[1] J. Alcalá-fdez, L. Sánchez, S. García, M.J.D. Jesus, S. Ventura, J.M. Garrell, J. Otero, J. Bacardit, V.M. Rivas, J.C. Fernández, F. Herrera, Keel: a software tool to assess evolutionary algorithms for data mining problems, Soft Computing - A Fusion of Foundations, Methodologies and Applications 13 (3) (2008) 307–318.
[2] AmericanCancerSociety. How many men get prostate cancer? <http://www.cancer.org/docroot/CRI/content/CRI_2_2_1X_How_many_men_get_prostate_cancer_36.asp>.
[3] A. Arcuri, X. Yao, Co-evolutionary automatic programming for software development, Information Sciences (2010), in press, http://dx.doi.org/10.1016/j.ins.2009.12.019.
[4] A. Asuncion, D. Newman, UCI machine learning repository (2007).

[5] W. Banzhaf, F.D. Francone, P. Nordin, The effect of extensive use of the mutation operator on generalization in genetic programming using sparse data sets, in: In Parallel Problem Solving from Nature IV, Proceedings of the International Conference on Evolutionary Computation, Springer Verlag, 1996, pp. 300–309.
[6] F. Berlanga, A. Rivera, M. del Jesus, F. Herrera, GP-COACH: genetic programming-based learning of compact and accurate fuzzy rule-based classification systems for high-dimensional problems, Information Sciences 180 (8) (2010) 1183–1200.
[7] A.S. Bickel, R.W. Bickel, Tree structured rules in genetic algorithms, In Proceedings of the Second International Conference on Genetic Algorithms and their Application, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1987.
[8] M.C.J. Bot, Feature extraction for the k-nearest neighbour classifier with genetic programming, In EuroGP '01: Proceedings of the Fourth European Conference on Genetic Programming, Springer-Verlag, London, UK, 2001.
[9] D.A. Cieslak, N.V. Chawla, A framework for monitoring classifiers' performance: when and why failure occurs?, Knowledge and Information Systems 18 (1) (2009) 83–108
[10] N.L. Cramer, A representation for the adaptive generation of simple sequential programs, In Proceedings of the 1st International Conference on Genetic Algorithms, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1985.
[11] C. Darwin, On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life, John Murray, London, UK, 1859.
[12] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[13] P.G. Espejo, S. Ventura, F. Herrera, A survey on the application of genetic programming to classification, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 40 (2) (2010) 121–144.
[14] M. Evett, T. Fernandez, Numeric mutation improves the discovery of numeric constants in genetic programming, in: J. Koza (Ed.), Proceedings of the Third Annual Genetic Programming Conference, Morgan Kaufmann, Madison, WI, 1998, pp. 66–71.
[15] D.C. Fernandez, R. Bhargava, S.M. Hewitt, I.W. Levin, Infrared spectroscopic imaging for histopathologic recognition, Nature Biotechnology 23 (4) (2005) 469–474.
[16] C. Fujiko, J. Dickinson, Using the genetic algorithm to generate lisp source code to solve the prisoner's dilemma, in: Proceedings of the Second International Conference on Genetic Algorithms and their application, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1987, pp. 236–240.
[17] C. Gagné, M. Parizeau, Genericity in evolutionary computation software tools: principles and case study, International Journal on Artificial Intelligence Tools 15 (2) (2006) 173–194.
[18] S. García, A. Fernández, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, Soft Computing 13 (10) (2009) 959–977.
[19] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Information Sciences 180 (10) (2010) 2044–2064.
[20] S. García, F. Herrera, An extension on 'statistical comparisons of classifiers over multiple data sets' for all pairwise comparisons, Journal of Machine Learning Research 9 (2008) 2677–2694.
[21] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[22] A. Guérin-Dugué et al. Deliverable R3-B1-P - Task B1: Databases. Technical report, Elena-NervesII "Enhanced Learning for Evolutive Neural Architecture, ESPRIT-Basic Research Project Number 6891, June 1995, Anonymous FTP:/pub/neural-nets/ELENA/Databases.ps.Z on ftp.dice.ucl.ac.be.
[23] H. Guo, A.K. Nandi, Breast cancer diagnosis using genetic programming generated feature, Pattern Recognition 39 (5) (2006) 980–987.
[24] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.
[25] I. Guyon, S. Gunn, M. Nikravesh, L. Zadeh (Eds.), Feature Extraction, Foundations and Applications, Springer, 2006.
[26] C. Harris, An investigation into the Application of Genetic Programming techniques to Signal Analysis and Feature Detection, PhD thesis, University College, London, 26 Sept. 1997.
[27] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, USA, 1975.
[28] K.-A. Kim, S.-Y. Oh, H.-C. Choi, Facial feature extraction using pca and wavelet multi-resolution images, in: Sixth IEEE International Conference on Automatic Face and Gesture Recognition, IEEE Computer Society, Los Alamitos, CA, USA, 2004, p. 439.
[29] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1995, pp. 1137–1143.
[30] M. Kotani, S. Ozawa, M. Nakai, K. Akazawa, Emergence of feature extraction function using genetic programming, In KES (1999) 49–152.
[31] P. Kouchakpour, A. Zaknich, T. Bräunl, Population variation in genetic programming, Information Sciences 177 (17) (2007) 3438–3452.
[32] P. Kouchakpour, A. Zaknich, T. Bräunl, Dynamic population variation in genetic programming, Information Sciences 179 (8) (2009) 1078–1091.
[33] J. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, The MIT Press, Cambridge, MA, 1992.
[34] J. Koza, Genetic programming II: Automatic Discovery of Reusable Programs, Complex Adaptive Systems, MIT Press, Cambridge, Mass, 1994.
[35] J.R. Koza, M.J. Streeter, M.A. Keane, Routine high-return human-competitive automated problem-solving by means of genetic programming, Information Sciences 178 (23) (2008) 4434–4452.
[36] I.W. Levin, R. Bhargava, Fourier transform infrared vibrational spectroscopic imaging: integrating microscopy and molecular recognition, Annual Review of Physical Chemistry 56 (2005) 429–474.
[37] J.-Y. Lin, H.-R. Ke, B.-C. Chien, W.-P. Yang, Classifier design with feature selection and feature extraction using layered genetic programming, Expert Systems with Applications 34 (2) (2008) 1384–1393.
[38] H. Liu, H. Motoda, Feature Extraction, Construction and Selection: A Data Mining Perspective, Kluwer Academic, Boston, 1998. vol. SECS 453.
[39] X. Llorà, J.M. Garrell, Knowledge-independent data mining with fine-grained parallel evolutionary algorithms, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001), Morgan Kaufmann Publishers, 2001, pp. 461–468.
[40] X. Llorà, A. Priya, R. Bhargava, Observer-invariant histopathology using genetics-based machine learning, Natural Computing: An International Journal 8 (1) (2009) 101–120.
[41] X. Llorà, R. Reddy, B. Matesic, R. Bhargava, Towards better than human capability in diagnosing prostate cancer using infrared spectroscopic imaging, in: GECCO '07: Proceedings of the Ninth Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2007, pp. 2098–2105.
[42] U.-M. O'Reilly, An Analysis of Genetic Programming, PhD thesis, Carleton University, Ottawa-Carleton Institute for Computer Science, Ottawa, Ontario, Canada, 1995.
[43] M. Pei, E.D. Goodman, W.F. Punch. Pattern discovery from data using genetic algorithms, in: Proceeding of First Pacific-Asia Conference Knowledge Discovery & Data Mining(PAKDD-97), 1997.
[44] A. Piszcz, T. Soule, A survey of mutation techniques in genetic programming, in: GECCO '06: Proceedings of the Eigth Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2006, pp. 951–952.
[45] I.T. Podolak, Facial component extraction and face recognition with support vector machines, in: FGR '02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, IEEE Computer Society, Washington, DC, USA, 2002, p. 83.
[46] R. Poli, W.B. Langdon, N.F. Mcphee, A Field Guide to Genetic Programming, Lulu Enterprises Ltd, UK, 2008.
[47] J. Quiñonero Candela, M. Sugiyama, A. Schwaighofer, N.D. Lawrence, Dataset Shift in Machine Learning, The MIT Press, 2009.
[48] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
[49] C. Ryan, M. Keijzer, An analysis of diversity of constants of genetic programming, in: C. Ryan, T. Soule, M. Keijzer, E. Tsang, R. Poli, E. Costa (Eds.), Genetic Programming, Proceedings of EuroGP'2003, LNCS, vol. 2610, Springer-Verlag, Essex, 2003, pp. 404–413.
[50] J.R. Sherrah, R.E. Bogner, A. Bouzerdoum, The evolutionary pre-processor: automatic feature extraction for supervised classification using genetic programming, Proceedings of the Second International Conference on Genetic Programming, vol. GP-97, Morgan Kaufmann, 1997, pp. 304–312.
[51] D.J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, 4th ed., Chapman & Hall/CRC, 2007.

[52] M.G. Smith, L. Bull, Genetic programming with a genetic algorithm for feature construction and selection, Genetic Programming and Evolvable Machines 6 (3) (2005) 265–281.
[53] W.A. Tackett, Genetic programming for feature discovery and image discrimination, in: Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993, pp. 303–311.
[54] K. Wang, S. Zhou, C.A. Fu, J.X. Yu, F. Jeffrey, X. Yu, Mining changes of classification by correspondence tracing, in: Proceedings of the 2003 SIAM International Conference on Data Mining (SDM 2003), 2003.
[55] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin 1 (6) (1945) 80–83.
[56] N. Wyse, R. Dubes, A. Jain, A critical evaluation of intrinsic dimensionality algorithms a critical evaluation of intrinsic dimensionality algorithms, in: E.S. Gelsema, L.N. Kanal (Eds.), Pattern Recognition in Practice, Morgan Kauffman Publishers, Inc., Amsterdam, 1980, pp. 415–425.
[57] Y. Yang, X. Wu, X. Zhu, Conceptual equivalence for contrast mining in classification learning, Data & Knowledge Engineering 67 (3) (2008) 413–429.
[58] A. Zafra, S. Ventura, G3P-MI: a genetic programming algorithm for multiple instance learning, Information Sciences, 180 (23) (2010) 4496–4513.
[59] J.H. Zar, Biostatistical Analysis, 5th ed., Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2007.
[60] Y. Zhang, H. Li, M. Niranjan, P. Rockett, Applying cost-sensitive multiobjective genetic programming to feature extraction for spam E-mail filtering, in: Proceedings of the 11th European Conference on Genetic Programming, EuroGP 2008, Lecture Notes in Computer Science, vol. 4971, Springer, Naples, 2008, pp. 325–336.
[61] Y. Zhang, P.I. Rockett, A generic optimal feature extraction method using multiobjective genetic programming, Technical Report VIE 2006/001, Department of Electronic and Electrical Engineering, University of Sheffield, UK, 2006.
[62] Y. Zhang, P.I. Rockett, A generic multi-dimensional feature extraction method using multiobjective genetic programming, Evolutionary Computation 17 (1) (2009) 89–115.

3. Interactions between Dataset Shift and other classification issues: Focus on Imbalanced Datasets and k-fold Cross-Validation

The papers associated to this part are:

3.1. Study on the relationship between class imbalance and dataset shift regarding classifier performance

- J.G. Moreno-Torres, F. Herrera, A Preliminary Study on Overlapping and Data Fracture in Imbalanced Domains by means of Genetic Programming-based Feature Extraction, Proceedings of 10th International Conference on Intelligent Design and Applications (ISDA), 2010, pages 501-506.

  - Status: **Published**.
  - Conference ranking (CORE 2008): C.

A Preliminary Study on Overlapping and Data Fracture in Imbalanced Domains by means of Genetic Programming-based Feature Extraction

Jose G. Moreno-Torres, Francisco Herrera
Department of Computer Science and Artificial Intelligence
Universidad de Granada, 18071 Granada, Spain.
(jose.garcia.mt,herrera)@decsai.ugr.es

***Abstract*—The classification of imbalanced data is a well-studied topic in data mining. However, there is still a lack of understanding of the factors that make the problem difficult. In this work, we study the two main reasons that make the classification of imbalanced datasets complex: overlapping and data fracture. We present a Genetic Programming-based feature extraction method driven by Rough Set Theory to help visualize the data in a bidimensional graph, to better understand how the presence of overlapping and data fractures affect classification performance.**

***Keywords*-imbalanced data; overlapping; data fracture; feature extraction; genetic programming; rough set theory;**

I. Introduction

The classification of imbalanced data is a priority issue in the literature nowadays [1], [2]. Most of the approaches presented are based on preprocessing the data, whether it is by oversampling the minority class or undersampling the majority one. Excellent accuracy results have been obtained, but there is still room to improve.

This contribution does not seek to better the classification performance obtained by existing proposals, but rather to analyze the two main factors where the real complexity of the problems lies:

- Overlapping: The examples of the minority class share a region with the majority one, where all the examples are intertwined. This is a problem intrinsic to the data. This issue has been studied in [3], [4].
- Data Fracture: There is a change in data distribution between the training and test sets, often in the minority class. The incidence of this issue depends on the partitioning of the data. The problem of data fracture (or dataset shift, as some authors call it) is relatively new [5], [6], [7], [8], and we are not aware of any studies regarding imbalanced datasets published so far.

To help perform a visual analysis of the data, we propose the 'Genetic Programming-based feature extraction using Rough Set Theory' algorithm (GP-RST), which is based on the application of the GP paradigm [9] as a feature extraction tool, using RST [10] techniques to estimate the fitness of individuals. It obtains a transformation from the original feature space into a bidimensional one where the classes are as separated as possible; serving both as a visualization tool and as a competitive preprocessing technique for imbalanced datasets. GP-RST is more suitable for the visualization of imbalanced domains than other feature extraction techniques because the fitness is calculated for each class and then aggregated, being therefore independent of the class imbalance in the training set.

The application of GP-RST has permitted the discovery of three possible situations, which are all easily visualizable in the bidimensional feature space it extracts:

1 The dataset presents a low amount of overlapping and data fracture, resulting in a good behavior both in terms of training and test classifier performance.
2 The dataset presents a high amount of overlapping, resulting in a poor classifier performance both in training and test.
3 There is a significant amount of data fracture, which produces an overfitting issue leading to a big gap between training and test set performance.

This contribution is organized as follows: We begin with some notation specifications in section II. In section III we briefly introduce the relevant concepts of RST. Section IV includes a description of the GP-RST algorithm, while section V shows the experimental procedure and classifier performance results. Section VI presents a visual analysis in terms of overlap and data fracture. Lastly, some concluding remarks are made in section VII.

II. Notation

A classification problem is considered with:

- A set of input attributes $A = \{a_i/i = 1, ..., n_v\}$, where n_v is the number of features of the problem.
- A set of values for the target variable (class) $C = \{C_j/j = \{1, \cdots, n_c\}\}$, where n_c is the number of different values for the class variable.
- A set of examples $E = \{e^h = (e^h_1, ..., e^h_{n_v}, C^h)/h = 1, \cdots, n_e\}$, where C^h is the class label for the example e^h, and n_e is the number of examples.
- The range of a variable i is defined as $range_i = (e^m_i) - (e^n_i)/\forall h{:}(e^m_i >= e^h_i \ \& \ e^n_i <= e^h_i)$.
- The number of examples of class C_j in E is noted as $n_e^{C_j}$.

978-1-4244-8136-1/10/$26.00 ©2010 IEEE

When applying GP-RST to obtain new features,

- The set of new features is noted as $Y = \{y_1, y_2\}$,
- The new features are functional mappings of A, represented as

$$Y = \{f_1(A), f_2(A)/f_i(A) = f_i(a_1, \cdots, a_{n_v})\}$$

- The result of applying a function f_i to a sample e^h is denoted as $f_i(e^h) = f_i(e^h_1, \cdots, e^h_{n_v})$.
- $E^{'}$ results of applying f_1, f_2 to a set of examples E:

$$E^{'} = \{e^{'h} = (f_1(e^h), f_2(e^h), C^h)/h = 1, \cdots, n_e\}$$

III. Introduction to Rough Set Theory

This section includes the definition of the RST concepts that are relevant to this work. For an in-depth study of the topic, see [10].

- Information System and Decision System: Let a set of attributes $A = \{a_1, a_2, \ldots, a_{n_v}\}$ and a non-empty, finite set called the universe U, with instances described using the attributes a_i; Information System is the name given to the pair (U, A). If a new attribute d called decision is attached to each element of U, indicating the decision made in that state or situation, then a Decision System is created $(U, A \cup \{d\})$, where $d \notin A$ is the decision attribute.
- The attribute of decision d induces a partition of the object universe U. Let a set of integer numbers$\{1, \ldots, l\}$, $X_i = \{x \in U : d\,(x) = i\}$, then $\{X_1, \ldots X_l\}$ is a collection of equivalence classes, called decision classes, where two objects belong to the same class if they have the same decision attribute value. In the case of this contribution, d corresponds to the class variable.
- The novelty of the RST are the lower and upper approximations of a subset $X \subseteq U$. These concepts were originally introduced in reference to an indiscernibility relation R. In classical RST, R is defined as an equivalence relation. This approach is extended by accepting that objects that are not indiscernible but sufficiently close or similar can be grouped into the same class. The aim is to construct a similarity relation $R^{'}$ from the indiscernibility relation R by relaxing the original conditions for indiscernibility.
- The similarity relation used in this work is defined as

$$R^{'}(x, y) = \begin{cases} 1 & \forall i(|x_i - y_i| < 0.1 * range_i) \\ 0 & otherwise \end{cases} \quad (1)$$

- The approximation of the set $X \subset U$, using the similarity relation $R^{'}$, has been induced as a pair of sets called lower approximation of X and upper approximation of X. The lower approximation $B_*(X)$ of X is defined as shown in equation 2.

$$B_*(X) = \{x \in X : R^{'}(x) \subseteq X\} \quad (2)$$

- Within RST, the meaning of the lower approximation of a decision system is of great interest for the analysis of new feature spaces. It consists of the objects that with absolute certainty belong to one class or another, guaranteeing that these instances are free of noise.
- Taking into account the equation defined in 2, the quality of the approximation of X is defined for the relation $R^{'}$ as:

$$\gamma(X) = \frac{|B_*(X)|}{|X|} \quad (3)$$

IV. A Genetic Programming-based feature extraction method driven by Rough Set Theory(GP-RST)

In this section we first present a formal expression of the problem at hand in subsection IV-A, followed by a general description of the GP-RST method in subsection IV-B, and we finish with a detailed explanation of the fitness calculation procedure in subsection IV-C.

A. Formal definition of the problem

The problem we are attempting to solve is, given a classification problem with a set of attributes A, and a set of examples E, obtain $f_1(A)$ and $f_2(A)$ such that $fitness(f_1(E), f_2(E))$ is maximized. The fitness calculation is based on the estimation of the separability between the classes through the maximization of the quality of approximation (Equation 3) for each class.

B. General description of GP-RST

Genetic Programming is an evolutionary computation technique that evolves expressions defined by a context-free grammar, by generating a starting population and applying crossover and mutation operators over it repeatedly, selecting on each generation the best potential solutions (expressions) according to a given fitness evaluation formula.

The GP-RST algorithm is a simple extension of a standard GP procedure with the following tweaks:

- It simultaneously evolves two trees, one for each dimension in the new feature space.
- It uses $\{x_1, ..., x_{n_v}, e\}$ as its terminal set, effectively evolving functional mappings of X.
- It uses $\{+, -, \times, \div\}$ as its function set.

C. Fitness evaluation

The fitness evaluation procedure, as has been expressed before, is based upon RST, more specifically it is associated to the quality of approximation of each of the classes.

V. Experimental Framework and Results

This section begins with a general description of the experimental procedure, followed by an enumeration of the datasets used in subsection V-A, then the specific parameters chosen for the experimentation can be seen in subsection

Algorithm 1 Fitness evaluation procedure

1. Obtain $E' = \{e'^h = (f_1(e^h), f_2(e^h), C^h)/h = 1, ..., n_e\}$, where f_1 and f_2 are the expressions encoded on each of the trees of the individual being evaluated.
2. For each class label $C_i \in C : i = 1, ..., n_c$,
   2.1 Build a rough set X_i containing all the elements of class C_i.
   2.2 Calculate the lower approximation of X_i, $B_*(X_i)$.
   2.3 The fitness of the chromosome for class C_i is estimated as the quality of the approximation over X_i, $\gamma(X_i)$.
3. The fitness of the chromosome is the geometric mean of the ones obtained for each class:
$$fitness = \sqrt[n_c]{\prod_{i=1}^{n_c} \gamma(X_i)}.$$

V-B. Finally, the classifier performance results are presented in subsection V-C.

The effectiveness of the preprocessing methods was measured in terms of classifier performance. Since the classical accuracy measures are not suitable to highly imbalanced domains, the performance was measured using the geometric mean of the accuracies per class [11]:

$$ClassifierPerformance = \sqrt{\frac{TP}{TP+FN} * \frac{TN}{TN+FP}} \quad (4)$$

where TP, TN, FP and FN stand for True Positives, True Negatives, False Positives and False Negatives respectively. The classifier used for all experiments was C4.5 [12], since it is a fast and efficient classifier that has been commonly used in the literature regarding imbalanced datasets. In any case, the choice of classifier does not have any influence in the visual analysis.

The testing procedure utilized was the standard in the literature, using a 5-fold cross validation technique where only the training set was used to do the preprocessing. We tested three different cases:

- The original dataset with no preprocessing, denoted as 'None'.
- The bidimensional dataset that results from applying GP-RST.
- SMOTE with ENN cleaning [13], a hybrid preprocessing method that first oversamples the minority class using SMOTE [14], and then cleans up the borders using the Edited Nearest Neighbor rule.

A schematic representation of the experimental procedure can be found in Figure 1. The GP implementation was based on the Open Beagle library [15], and we used the KEEL software [16] to carry out all the experiments and the statistical tests.

A. Datasets

The datasets used in this study were obtained from the KEEL dataset repository [17], which are in turn variations

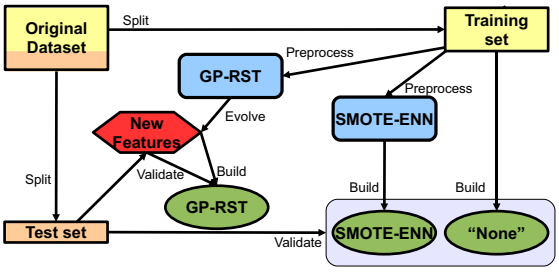


Figure 1. Schematic representation of the experimental procedure

of well known UCI datasets [18]. Table I presents the datasets, detailing the number of samples n_s, number of variables n_v and Imbalance Ratio (IR), which is calculated as $\frac{n_s^{majorityClass}}{n_s^{minorityClass}}$. All the datasets used are binary classification problems, but the GP-RST algorithm is capable of working with multiclass problems without any modifications.

Table I
DATASETS USED FOR THE EXPERIMENTAL STUDY

Dataset	IR	n_v	n_s
ecoli_0137v26	39.14	7	281
yeast_2v8	23.10	8	482
glass_5	22.78	9	214
shuttle_2v4	20.50	9	129
glass_016v5	19.44	9	184
pageblocks_13v4	15.86	10	472
ecoli_4	15.80	7	336
glass_4	15.46	9	214
yeast_1v7	14.30	7	459
glass_2	11.59	9	214
glass_016v2	10.29	9	192
yeast_2v4	9.08	8	516

B. Parameters

This subsection presents the parameter values chosen for the GP evolution. In this work, we decided not to squeeze the maximum performance from the method but to focus on the interpretation of the visual results, so most of the parameters were fixed to common default values.

The specific values for the parameters are presented in Table II.

C. Classifier performance results

This subsection presents the results obtained by the different preprocessed datasets, in terms of the test-set classifier performance (see Equation 4) obtained by the C4.5 classifier. They are shown in Table III.

To check whether the differences in performance are significant, we performed a statistical analysis of the results by means of a non-parametric test.

In [19], [20], [21], [22] a set of simple, safe and robust non-parametric tests for statistical comparisons of classifiers

Table II
EVOLUTIONARY PARAMETERS FOR THE GP-RST PROCEDURE

Parameter	Value
Number of trees	2
Population size	10000
Duration of the run	200 generations
Selection operator	Tournament, no replacement
Tournament size	3
Crossover operator	One-point crossover
Crossover probability	0.9
Mutation operator	Replacement & Swap
Replacement mutation prob	0.001
Swap mutation prob	0.01
Max depth, swapped-in subtree	5

Table III
CLASSIFIER PERFORMANCE RESULTS

	C4.5 performance		
Dataset	None	GP-RST	SMOTE-ENN
ecoli_0137v26	0.8436	0.8405	0.7462
yeast_2v8	0.2226	0.6635	0.7542
glass_5	0.8776	0.8798	0.9405
shuttle_2v4	0.9129	0.9877	1.0000
glass_016v5	0.7389	0.9320	0.9943
pageblocks_13v4	0.9989	0.9764	0.9989
ecoli_4	0.7985	0.8916	0.8563
glass_4	0.7228	0.8683	0.7746
yeast_1v7	0.5719	0.5464	0.4828
glass_2	0.2407	0.2394	0.6976
glass_016v2	0.0000	0.0000	0.5333
yeast_2v4	0.7921	0.7996	0.8770
Average	0.6434	0.7188	0.8046

are recommended. One of them is the Wilcoxon Signed-Ranks Test [23], [24], which is the test that we have selected to do the comparisons. A complete description of the Wilcoxon Signed-Ranks Test and other non-parametric tests for pairwise and multiple comparisons, together with software for their use, can be found in the website available at http://sci2s.ugr.es/sicidm/.

We evaluated the methods by performing all pairwise comparisons among them, including the option of not doing any preprocessing, denoted as 'None'. The results are presented in Table IV.

Table IV
WILCOXON SIGNED-RANKS TEST RESULTS

Comparison	R^+	R^-	p-value (two-tailed)
None v GP-RST	15	51	0.05372
None v SMOTE-ENN	13	53	0.0392
GP-RST v SMOTE-ENN	28	50	0.2005

From the results shown in Table IV, we can extract the following conclusions:

- Both GP-RST and SMOTE-ENN significantly outperform not doing anything.
- GP-RST performs slightly worse than SMOTE-ENN, but the difference is not statistically significant.

VI. GRAPHICAL ANALYSIS OF OVERLAPPING AND DATA FRACTURE

In this section we present a set of sample visualizations of the bidimensional datasets obtained by GP-RST.

A. Good behavior

Figure 2 shows a case where GP-RST succeeded in finding a bidimensional mapping of the original features in the ecoli4 dataset where both classes are easily separable in the training set, and such a separation generalizes well to the test set. This is the ideal case, one where a classifier performs very well, both in training and test.

B. Overlap

Figure 3 presents a case where, due to the complex overlap between classes in the original dataset, the GP-RST procedure was not successful in finding a bidimensional mapping where they were separable. The classifier performance on the preprocessed dataset was as bad as it was without preprocessing. This is the type of issue that was studied by [3], [4].

C. Data fracture

Figure 4 shows a case where partial success was achieved in the classification of the training set, but none of the examples in the test set belong to the area where the classes are separable.

This issue is the one we would like to raise awareness about. Even though most authors know about the overlap problem, the data fracture one is usually not considered, and needs to be taken into account when analyzing the performance of new methods in imbalanced domains.

VII. CONCLUSIONS

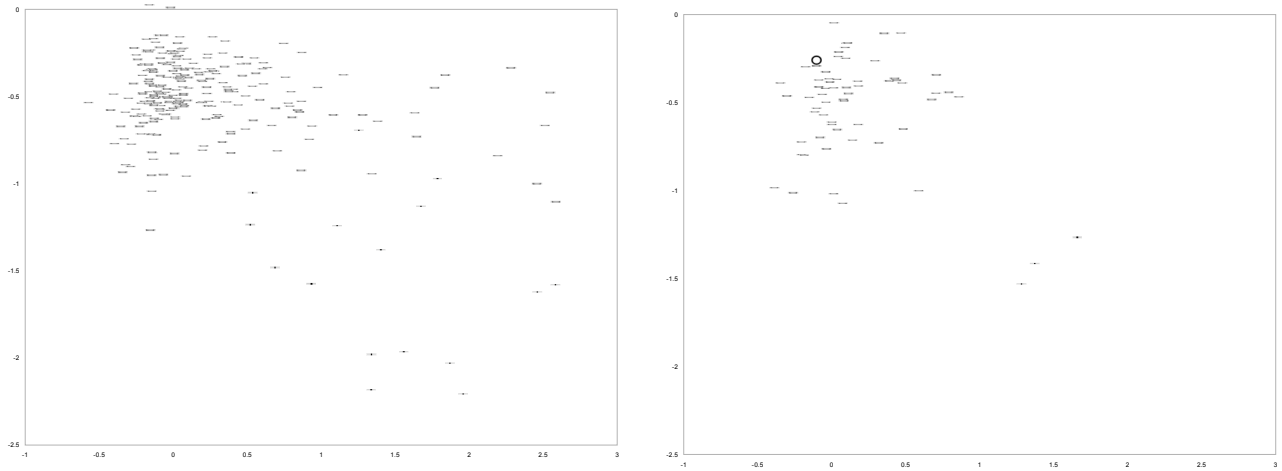
We have presented GP-RST, a GP-based feature extractor that employs RST techniques to estimate the fitness of individuals. We have shown GP-RST to be a competitive preprocessing method for highly imbalanced datasets, with the added advantage of providing bidimensional representations of the datasets it preprocesses, which are easily intepreted.

We have, through the analysis of the visual representations of the preprocessed datasets, observed a data fracture problem between training and test sets, specially in the minority class, that is affecting the classification performance.

We believe this discovery is very relevant since it challenges the usual assumptions when experimenting with preprocessing for highly imbalanced data. We intend to further study the issue, to test the hypothesis that data fracture is playing a major role in the complexity of classification in imbalanced domains.

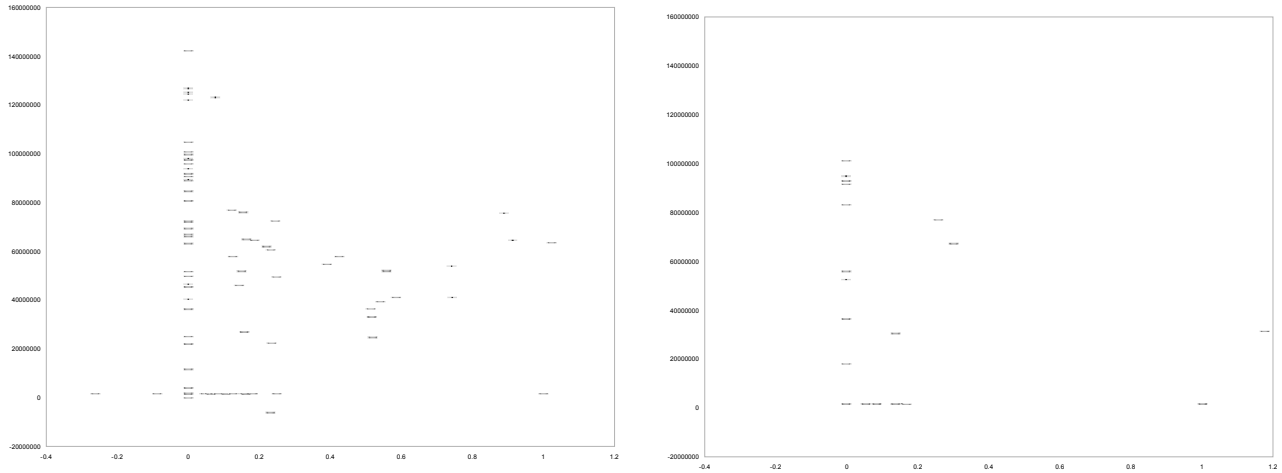
ACKNOWLEDGMENTS

Jose García Moreno-Torres is currently supported by a FPU grant from the Ministerio de Educación y Ciencia of the Spanish Government.



(a) Training set (0.9682) (b) Test set (0.9919)

Figure 2. Example of good behavior, dataset ecoli_4, 5^{th} partition. Classifier performance in parenthesis.



(a) Training set (0.3779) (b) Test set (0.0000)

Figure 3. Example of bad behavior by overlap, dataset glass_016v2, 4^{th} partition. Classifier performance in parenthesis.

REFERENCES

[1] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, September 2009.

[2] Y. M. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of Imbalanced Data: A Review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 4, pp. 687–719, 2009.

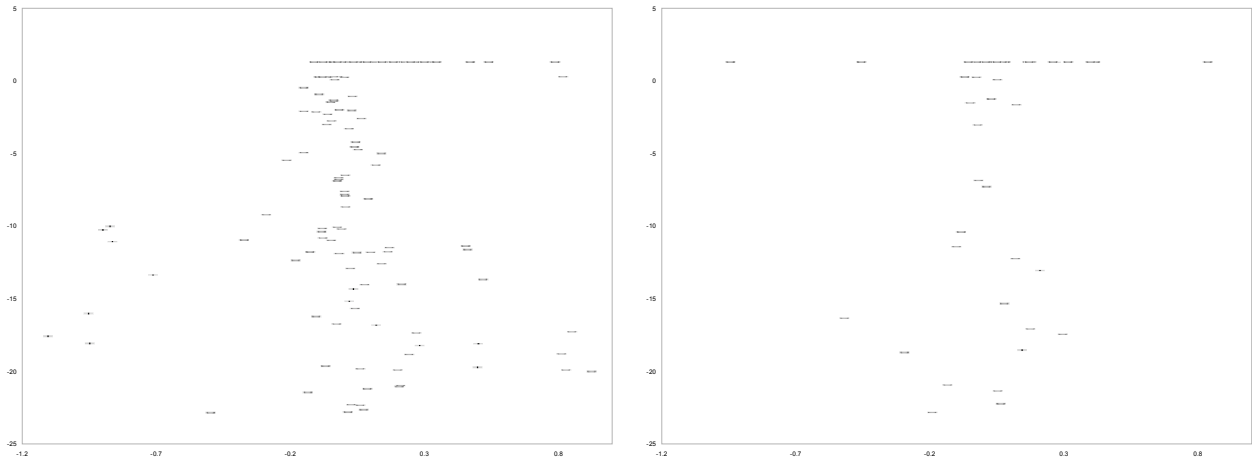
[3] V. García, R. A. Mollineda, and J. S. Sánchez, "On the k-NN performance in a challenging scenario of imbalance and overlapping," *Pattern Analysis & Applications*, vol. 11, no. 3-4, pp. 269–280, 2008.

[4] M. Denil and T. P. Trappenberg, "Overlap versus Imbalance," in *Canadian Conference on AI*, 2010, pp. 220–231.

[5] R. Alaiz-Rodríguez and N. Japkowicz, "Assessing the impact of changing environments on classifier performance," in *Canadian AI'08: Proceedings of the Canadian Society for computational studies of intelligence, 21st conference on Advances in artificial intelligence*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 13–24.

[6] J. Quiñonero Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[7] D. A. Cieslak and N. V. Chawla, "A framework for monitoring classifiers' performance: when and why failure occurs?" *Knowledge and Information Systems*, vol. 18, no. 1, pp. 83–108, 2009.



(a) Training set (0.6794) (b) Test set (0.0000)

Figure 4. Another example of bad behavior by data fracture, dataset glass_2, 2^{nd} partition. Classifier performance in parenthesis.

[8] J. G. Moreno-Torres, X. Llorà, D. E. Goldberg, and R. Bhargava, "Repairing Fractures between Data using Genetic Programming-based Feature Extraction: A Case Study in Cancer Diagnosis," *Information Sciences, In Press*, 2010.

[9] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: The MIT Press, 1992.

[10] Z. Pawlak, *Rough Sets. Theoretical Aspects of Reasoning about Data*. Dordrecht: Kluwer Academics, 1991.

[11] R. Barandela, J. S. Sánchez, V. García, and E. Rangel, "Strategies for learning in class imbalance problems," *Pattern Recognition*, vol. 36, no. 3, pp. 849–851, 2003.

[12] J. R. Quinlan, *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[13] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[15] C. Gagné and M. Parizeau, "Genericity in evolutionary computation software tools: Principles and case study," *International Journal on Artificial Intelligence Tools*, vol. 15, no. 2, pp. 173–194, 2006.

[16] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. D. Jesus, S. Ventura, J. M. Garrell, J. Otero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera, "Keel: A software tool to assess evolutionary algorithms for data mining problems," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 13, no. 3, pp. 307–318, 2009.

[17] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework," *Journal of Multiple-Valued Logic and Soft Computing, In Press*, 2010.

[18] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html

[19] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[20] S. García and F. Herrera, "An Extension on 'Statistical Comparisons of Classifiers over Multiple Data Sets' for all Pairwise Comparisons," *Journal of Machine Learning Research*, vol. 9, pp. 2677–2694, 2008.

[21] S. García, A. Fernández, J. Luengo, and F. Herrera, "A study of statistical techniques and performance measures for Genetics-Based Machine Learning: Accuracy and Interpretability," *Soft Computing*, vol. 13, no. 10, pp. 959–977, 2009.

[22] ——, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.

[23] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[24] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures (4th Edition)*. Chapman & Hall/CRC, 2007.

3.2. Study on the impact of partition-induced dataset shift on k-fold cross-validation

- J.G. Moreno-Torres, J.A. Sáez, F. Herrera, Study on the impact of partition-induced dataset shift on k-fold cross-validation. IEEE Transactions on Neural Networks and Learning Systems, 23:8 (2012) 1304-1312. doi: 10.1109/TNNLS.2012.2199516.

  - Status: **Published**.
  - Impact Factor (JCR 2011): 2.952.
  - Subject Category: Computer Science, Artificial Intelligence. Ranking 12 / 111 (Q1).
  - Subject Category: Computer Science, Hardware & Architecture. Ranking 1 / 50 (Q1).
  - Subject Category: Computer Science, Theory & Methods. Ranking 4 / 99 (Q1).
  - Subject Category: Engineering, Electrical & Electronic. Ranking 19 / 245 (Q1).

Study on the Impact of Partition-Induced Dataset Shift on k-fold Cross-Validation

Jose García Moreno-Torres, José A. Sáez, and Francisco Herrera, *Member, IEEE*

***Abstract*— Cross-validation is a very commonly employed technique used to evaluate classifier performance. However, it can potentially introduce dataset shift, a harmful factor that is often not taken into account and can result in inaccurate performance estimation. This paper analyzes the prevalence and impact of partition-induced covariate shift on different k-fold cross-validation schemes. From the experimental results obtained, we conclude that the degree of partition-induced covariate shift depends on the cross-validation scheme considered. In this way, worse schemes may harm the correctness of a single-classifier performance estimation and also increase the needed number of repetitions of cross-validation to reach a stable performance estimation.**

***Index Terms*— Covariate shift, cross-validation, dataset shift, partitioning.**

I. Introduction

In order to evaluate the expected performance of a classifier over a dataset, k-fold cross-validation schemes are commonly used in the classification literature [1]. Also, when comparing classifiers, it is common to compare them according to their performances averaged over a number of iterations of cross-validation. Even though it has been proved that these schemes asymptotically converge to a stable value, which allows realistic comparisons between classifiers [2], [3], in practice a very low number of iterations are often used. The most common variations are 2×5, 5×2, and 10×1, with this notation meaning 2-folds iterated five times, 5-folds iterated two times, and 10-folds iterated once, respectively. Note that when more than one iteration takes place, the partitions are assumed to be constructed independently.

The topic of data stability and classifier bias is very relevant to the field, as can be seen in the numerous attempts to design unbiased classifiers in the recent literature [4], [5], or in the recent research on streaming data [6] . While those designs are definitely worthwhile, we believe that a study of the intrinsic characteristics of the data is needed to have a full picture of the problem. Among the said data characteristics, the amount of partition-induced dataset shift is very relevant and, to the best of our knowledge, usually not taken into account. Here, we try to prove the relevance and need for accounting of this particular issue.

This paper studies the intrinsic variability present in k-fold cross-validation schemes from the point of view of dataset shift [7], [8], which is defined as the situation where the data the classifier is trained on and the data the classifier is going to be used on do not follow the same distribution. More specifically, we focus on covariate shift (a specific kind of dataset shift where the covariates follow a different distribution in the training and test datasets), and the situations where it may appear and cause inaccurate classifier performance estimations.

This paper analyzes how different partitioning methods can introduce dataset shift (or, more specifically, covariate shift) and the effect it has over both the reliability of the estimation of a classifier performance based on a low number of iterations of k-fold cross-validation, and the number of iterations needed to reach a stable classifier performance estimation.

To best analyze the impact of dataset shift, we use four different strategies to create the partitioning.

1) *Standard stratified cross-validation (SCV)*, which is the most commonly employed method in the literature. It places an equal number of samples of each class on each partition to maintain class distributions equal in all partitions. For an example of its use, see [9].
2) *Distribution-balanced SCV (DB-SCV)* [10], a method that attempts to minimize covariate shift by keeping data distribution as similar as possible between training and test folds by maximizing diversity on each fold and trying to keep all folds as similar as possible to each other.
3) *Distribution optimally balanced SCV (DOB-SCV)*, a slight modification of the above and an original contribution of the work presented here, tries to improve the performance of DB-SCV by taking into account more information when choosing in which fold to place each sample.
4) *Maximally shifted SCV (MS-SCV)*, a method designed for testing the maximal influence partition-based covariate shift can have on classifier performance by introducing the maximum possible amount of shift on each partition. To do so, it does the opposite as DB-SCV and creates folds that are as different as possible to each other.

While there is a published work that proposes a different cross-validation strategy [11] designed specifically to combat covariate shift, we chose not to include it in this paper

Manuscript received November 23, 2011; revised May 2, 2012; accepted May 3, 2012. Date of publication June 26, 2012; date of current version July 16, 2012. This work was supported in part by Project TIN2011-28488 and Project P10-TIC-6858. The work of J. G. Moreno-Torres and J. A. Sáez was supported by FPU grants from the Ministerio de Educación y Ciencia of the Spanish Government.

The authors are with the Department of Computer Science and Artificial Intelligence, University of Granada, Granada 18001, Spain (e-mail: jose.garcia.mt@decsai.ugr.es; smja@decsai.ugr.es; herrera@decsai.ugr.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2012.2199516

2162–237X/$31.00 © 2012 IEEE

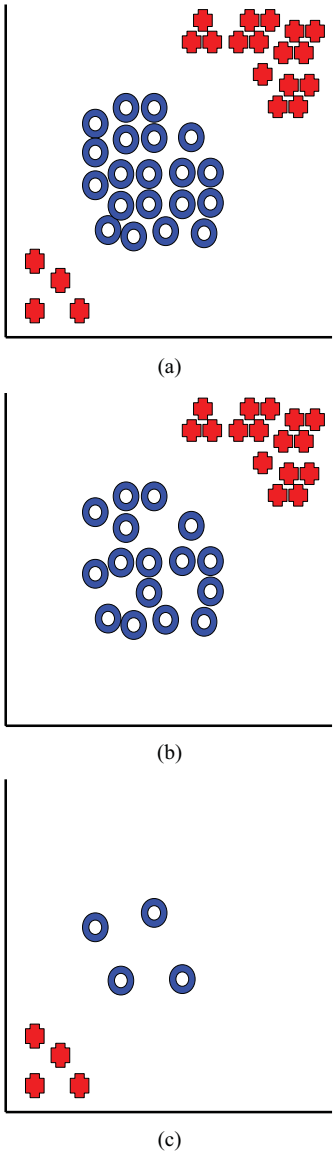


Fig. 1. Extreme example of partition-based covariate shift. Note that the examples on the bottom left of the "cross" class will be wrongly classified because of covariate shift. (a) Full dataset. (b) Training set. (c) Test set.

because it is designed to train classifiers in problems where covariate shift is already present, while the intent here is to analyze to what extent general-purpose cross-validation strategies generate extra covariate shift which is not intrinsic to the problem.

The goal of this paper is to analyze the accuracy of classifier performance prediction both when a low number of iterations of k-fold cross-validation are used and when enough of them are used so that a stable value has been achieved. More specifically, we study the following.

1) The accuracy of a single cross-validation experiment in terms of 1 v 1 classifier comparison, and whether different partitioning methods can have an impact on it.
2) The number of independent cross-validation experiments necessary to converge to a stable result in terms of 1 v 1 classifier comparison, also analyzing whether different partitioning methods produce different results.

Algorithm 1 SCV Partitioning Method

```
for each class c_j ∈ C do
  n ← count(c_j)/k
  for each fold F_i (i = 0, ..., k − 1) do
    E ← randomly select n examples of class c_j from D
    F_i ← F_i ∪ E
    D ← D \ E
  end for
end for
```

A supplementary material website has been created for this paper, which can be found at http://sci2s.ugr.es/covariate-shift-cross-validation.

The remainder of this paper is organized as follows. Section II provides a background on cross-validation and dataset shift. In Section III, the different partitioning methods used for the experimentation in this paper are detailed. Section IV shows the datasets and classification algorithms used in the experimental study. Section V shows the strategy employed to test the suitability of each partitioning method, while Section VI shows the results obtained. This paper is then closed with a few concluding remarks and recommendations in Section VII.

II. Background

This section presents a brief introduction to classifier evaluation through cross-validation in Section II-A and to dataset shift in Section II-B, introducing the concepts relevant to this paper.

A. Cross-Validation for Classifier Evaluation

Cross-validation is a technique used for assessing how a classifier will perform when classifying new instances of the task at hand. One iteration of cross-validation involves partitioning a sample of data into two complementary subsets: training the classifier on one subset (called the training set) and testing its performance on the other subset (test set).

In k-fold cross-validation, the original sample is randomly partitioned into k subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the classifier, and the remaining $k-1$ subsamples are used as training data. The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as the test data. The k results from the folds are then averaged to produce a single performance estimation.

Cross-validation has been the subject of profuse study in the literature, some of the most interesting and relevant results are listed here:

1) repeated iterations of cross-validation asymptotically converge to a correct estimation of classifier performance [2];
2) ten-fold cross-validation is better than leave-one-out validation for model selection, and also better than other k-fold options [1];
3) k-fold cross-validation tends to underestimate classifier performance [1].

Algorithm 2 DB-SCV Partitioning Method

for each class $c_j \in C$ **do**
  $e \leftarrow$ randomly select an example of class c_j from D
  $i = 0$
  while count$(c_j) > 0$ **do**
    $F_i \leftarrow F_i \cup \{e\}$
    $D \leftarrow D \setminus \{e\}$
    $i = (i + 1) \bmod k$
    $e \leftarrow$ closest example to e of class c_j from D
  end while
end for

Algorithm 3 DOB-SCV Partitioning Method

for each class $c_j \in C$ **do**
  while count$(c_j) > 0$ **do**
    $e_0 \leftarrow$ randomly select an example of class c_j from D
    $e_i \leftarrow i$th closest example to e_0 of class c_j from D $(i = 1, \ldots, k-1)$
    $F_i \leftarrow F_i \cup \{e_i\}$ $(i = 0, \ldots, k-1)$
    $D \leftarrow D \setminus \{e_i\}$ $(i = 0, \ldots, k-1)$
  end while
end for

B. Dataset Shift

The term "dataset shift" refers to the issue where training and test data follow different data distributions [7], [8]. It can happen because of the intrinsic nature of the problem (for example, a classifier trained over financial data from the past five years and used to predict future market changes), or it can be introduced in cross-validation schemes without noticing.

This paper focuses on the latter, studying k-fold cross-validation strategies and the types and impact of dataset shift in them. There are two potential types of dataset shift.

1) *Prior Probability Shift:* It happens when the class distribution is different between the training and test sets [12]. In the most extreme example, the training set would not have a single example of a class, leading to a degenerate classifier. The problems caused by this kind of shift have already been studied, and it is commonly prevented by applying a SCV scheme [13].
2) *Covariate Shift:* In this case, it is the inputs that have different distributions between the training and test sets [14]. Fig. 1 depicts an extreme example of this type of shift that can also lead to extremely poor classifier performance. This type of shift is often ignored in the literature, and the analysis of its prevalence and potential impact is the main contribution of this paper.

III. Partitioning Methods

This section presents a detailed explanation of the different partitioning methods used for testing in this paper, including the pseudo-code to make the replication of our experiments easier. Some assumptions made throughout the pseudo-codes are as follows.

1) The number of folds in a given cross-validation implementation is denoted as k.
2) Folds are named $F_i (i = 0, \ldots, k-1)$. They are treated as a set of examples, and are initially empty.
3) D is another set of examples, initially containing all the examples in the dataset.
4) There is a set of classes $C = \{c_1, \ldots, c_m\}$, where m is the number of classes.
5) There is a function count(c_i) that returns the number of examples of class c_i in D.
6) These methods detail the way to construct the test sets; the training sets are simply the remainder of the dataset.

This section also includes, in Section III-E, an analysis of the differences between DB-SCV and DOB-SCV.

A. SCV

This is the standard method most authors in the field of classification apply. A pseudo-code explaining how it works can be seen in Algorithm 1.

SCV is a simple method: it counts how many samples of each class there are on the dataset, and distributes them evenly on the folds, so that each fold contains the same number of examples of each class. This avoids prior probability shift, since if there is an equal distribution class-wise on each fold, training and test set will have the same class distribution. However, this method does not take into account the covariates of the samples, so it can potentially generate covariate shift.

B. DB-SCV

This method, proposed in [10], adds an extra consideration to the partitioning strategy as an attempt to reduce covariate shift on top of preventing prior probability shift. The method follows the steps detailed in Algorithm 2.

The idea is that by assigning close-by examples to different folds, each fold will end up with enough representatives of every region, thus avoiding covariate shift. To achieve this goal, DB-SCV starts on a random unassigned example and assigns it to the first fold. It then hops to the nearest unassigned neighbor of the same class, and assigns it to the second fold, repeating the process until there are no more examples of that class (when it gets to the last fold, cycles and continues with the first one again). The whole process is repeated for each class.

C. DOB-SCV

This method includes a variation from the one above, and is an original contribution of the work described in this paper. Its basic difference with DB-SCV lies in the order in which the examples are picked to be assigned to each fold. The specifics about this method can be found in Algorithm 3.

Instead of choosing samples one by one like DB-SCV does, DOB-SCV picks a random unassigned example, and then finds its $k-1$ nearest unassigned neighbors of the same class. Once it has found them, it assigns each of those examples to a different fold. The process is repeated until all examples are assigned.

Algorithm 4 MS-SCV Partitioning Method

```
for each class c_j ∈ C do
  n ← count(c_j)/k
  e ← randomly select an example of class c_j from D
  for each fold F_i (i = 0, ..., k − 1) do
    for s = 1 → n do
      F_i ← F_i ∪ {e}
      D ← D \ {e}
      e ← closest example to e of class c_j from D
    end for
  end for
end for
```

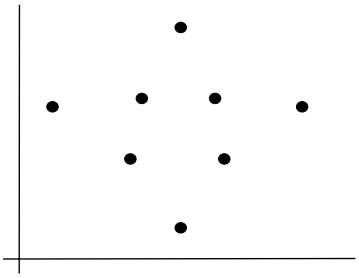


Fig. 2. Artificial dataset used to show differences between DB-SCV and DOB-SCV.

D. MS-SCV

This method is basically the opposite of the previous one in terms of covariate shift, trying to maximize it while keeping prior probability shift at a minimum. Its pseudo-code is shown in Algorithm 4.

MS-SCV is basically a mirrored version of DB-SCV: it picks an unassigned example at random, assigns it to a fold, and finds the nearest unassigned neighbor of the same class. However, instead of assigning it to the next fold, it assigns it to the same fold, and keeps assigning examples to the same fold until the maximum number of examples of that class have been assigned to the fold. Once that fold is "full," it goes to the next fold and repeats the process until all folds are filled. This procedure is again repeated for each class present in the dataset. In this case, the assignation of all close examples to the same fold creates incidences of severe covariate shift, since entire regions are kept without a single example representing them in some folds.

E. Difference Between DB-SCV and DOB-SCV

DB-SCV and DOB-SCV are similar methods with the same philosophy: they attempt to minimize covariate shift by distributing samples of the same class as evenly as possible in terms of their covariates. However, the way DB-SCV is designed makes it a little more sensitive to random choices, since the order in which it traverses the dataset depends only on the nearest neighbor each time, which, if the dataset is particularly poorly suited for this method, can lead to bad performance, while DOB-SCV is more resilient to this factor because of it restarting its exploration of the dataset more often and exploring several directions at once.

To illustrate the type of situation where DB-SCV could perform worse than DB-SCV, we have designed an artificial dataset which can be seen in Fig. 2. This dataset was built to clearly show the situation, since the visualization of high-dimensional datasets is not straightforward, and thus real-world datasets are less suitable for this task. To simplify, let all the samples of this dataset be of the same class and focus only on the avoidance of covariate shift. Also, assume for simplicity a fourfold partitioning scheme, so considering there are eight samples and two will be assigned to each fold.

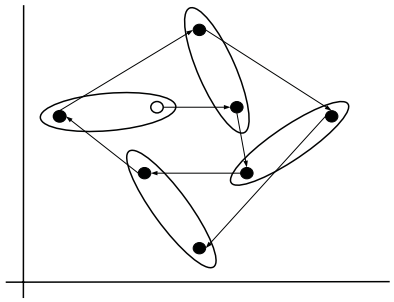


Fig. 3. Artificial dataset partitioned with DB-SCV. Arrows represent unassigned nearest neighbor exploration, white node is randomly chosen as starting point, and ellipses contain the partitions created.

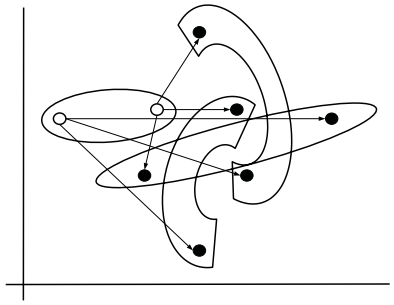


Fig. 4. Artificial dataset partitioned with DOB-SCV. Arrows represent unassigned nearest neighbor exploration, white nodes are randomly chosen as starting points, and shapes contain the partitions created.

In Fig. 3 the result of applying DB-SCV to the artificial dataset is shown. In it, arrows represent unassigned nearest neighbor exploration, white node is randomly chosen as starting point, and ellipses contain the partitions created. It can be seen how exploring only the nearest neighbor can lead the process to spiral around the center, resulting in a poor assignation of samples to folds and introduces a significant amount of covariate shift.

Fig. 4 shows the application of DOB-SCV to the same dataset, with the same starting point. It can be seen how DOB-SCV mostly avoids the pitfall because of its ability to see several neighbors at once, avoiding tunnel vision which can be costly in situations like this. The second white node in the figure corresponds to the second random choice, and we picked the one that results in the worst partitioning. The figure shows a better behavior for DOB-SCV than the one presented by DB-SCV, since the partitions created are better distributed in the domain space.

IV. Datasets and Classifiers

In this section, the experimental framework is presented showing the datasets used in Section IV-A and the classification algorithms studied in Section IV-B.

TABLE I
DATASETS USED FOR THE EXPERIMENTAL STUDY

Dataset	No. of attributes (R/I/N)	No. of examples
Appendicitis	7 (7/0/0)	106
Australian	14 (3/5/6)	690
Banana	2 (2/0/0)	5300
Bands	19 (13/6/0)	365
Breast	9 (0/0/9)	277
Bupa	6 (1/5/0)	345
Chess	36 (0/0/36)	3196
Crx	15 (3/3/9)	653
German	20 (0/7/13)	1000
Haberman	3 (0/3/0)	306
Heart	13 (1/12/0)	270
Hepatitis	19 (2/17/0)	80
Housevotes	16 (0/0/16)	232
Ionosphere	33 (32/1/0)	351
Mammographic	5 (0/5/0)	830
Monk-2	6 (0/6/0)	432
Mushroom	22 (0/0/22)	5644
Phoneme	5 (5/0/0)	5404
Pima	8 (8/0/0)	768
Saheart	9 (5/3/1)	462
Sonar	3 (60/0/0)	208
Spambase	57 (57/0/0)	4597
Spectfheart	44 (0/44/0)	267
Tic-tac-toe	9 (0/0/9)	958
Titanic	3 (3/0/0)	2201
Wdbc	30 (30/0/0)	569
Wisconsin	9 (0/9/0)	683

In order to achieve relevant results, 27 datasets and 9 classifiers were used. They can be seen in Sections IV-A and IV-B, respectively. All classifiers were compared against each other (resulting in 36 unique pairs) over their performance in the test set. The performance metric chosen was the area under the curve (AUC) [15], since it is less sensitive to imbalance than other commonly employed metrics, such as accuracy, and therefore allows us to obtain more solid conclusions.

A. Datasets

As has been mentioned before, we employed 27 datasets in our experimentation. They are all binary classification problems, and were obtained from the KEEL dataset repository [16]. When there were missing values, the whole example was eliminated. A list of the datasets used can be seen in Table I, where "(R/I/N)" refers to real, integer, and nominal attributes.

B. Classification Algorithms

Table II shows the nine classification algorithms employed in this paper, which were chosen to provide a wide range of classifiers. The parameters used were the default ones present in the KEEL tool [25], and are the ones suggested by the original authors of the methods.

TABLE II
CLASSIFICATION ALGORITHMS USED

Algorithm	Abbreviation	Type of classifier
Nearest neighbor $k = 1$ [17]	1 NN	Lazy learner
Nearest neighbor $k = 3$ [17]	3 NN	Lazy learner
C4.5 [18]	C4.5	Decision tree
Fuzzy unordered rule induction algorithm [19]	FURIA	Fuzzy rule-based (Mamdami)
Linear discriminant analysis [20]	LDA	Statistical
PART [21]	PART	Partial decision tree
Positive definite fuzzy classifier [22]	PDFC	Fuzzy rule-based (TSK)
Repeated incremental pruning to produce error reduction [23]	RIPPER	Rule-based
Support vector machine [24]	SVM	Support vector machine

V. ANALYZING PARTITIONING METHODS

We performed three independent experiments using the same procedure, where the only difference was the type of cross-validation scheme used. We tested the 2×5, 5×2, and 10×1 cross-validation schemes.

For each of the above validation schemes, we created 100 independent experiments using each of the methods described in Section III to test both single-experiment accuracy and number of iterations needed to converge to a stable result.

To evaluate single-iteration accuracy, we used the following procedure. Since the procedure is not trivial to understand, we include an example here with the case of 5×2 experiments; 10×1 and 2×5 are done analogously.

1) A reference is needed in order to know whether a classifier performance estimation is accurate. The said reference is based on the "true" performance of the classifiers. To obtain this "truth," we created an extra 200 independent partitions using SCV, and then averaged the performance of each classifier on each dataset over those 200 partitions. The performance is measured as AUC in the test set. The results can be seen in Table III.
2) Perform a Wilcoxon signed-ranks test with the averaged data for each classifier pair. The results can be seen in Table IV, where the numbers on each cell should be read as R+/R−/p-value (where R+ corresponds to row winning, and R− to column). Discard the classifier pairs where p-value > 0.1. In the table, the cases with p-value under 0.1 are marked in bold. We chose to focus only on the pairwise comparisons between classifiers where the true comparison showed a significant difference between classifiers, since it is harder to reach relevant conclusions from the cases where a significant difference could not be found.
3) For each of the 100 instances of 5×2 cross-validation created with each partitioning method, perform a Wilcoxon signed-ranks test for each classifier pair that

TABLE III
AVERAGED "TRUE" CLASSIFIER PERFORMANCE (AUC IN TEST SET)

Dataset	1 NN	3 NN	C45	FURIA	LDA	PART	PDFC	RIPPER	SVM
Appendicitis	0.7506	0.7450	0.7054	0.7265	0.7323	0.7028	0.7278	0.7305	0.6744
Australian	0.8228	0.8474	0.8449	0.8579	0.8649	0.6443	0.8262	0.8206	0.8045
Banana	0.8695	0.8822	0.8855	0.8780	0.5171	0.5617	0.8942	0.6637	0.9004
Bands	0.6904	0.6632	0.6211	0.6047	0.6021	0.5115	0.7068	0.6113	0.7060
Breast	0.5827	0.5842	0.5965	0.6255	0.6115	0.5049	0.6433	0.6151	0.5904
Bupa	0.6050	0.6266	0.6310	0.6554	0.6579	0.5206	0.6914	0.6349	0.6825
Chess	0.9692	0.9692	0.9930	0.9931	0.8510	0.8218	0.9955	0.9926	0.9839
Crx	0.8189	0.8542	0.8539	0.8653	0.5000	0.5561	0.8277	0.8272	0.8035
German	0.6275	0.6349	0.6303	0.6070	0.6438	0.5000	0.6490	0.6434	0.7056
Haberman	0.5462	0.5501	0.5745	0.5864	0.5637	0.5015	0.5575	0.5970	0.5564
Heart	0.7681	0.8038	0.7809	0.7970	0.8365	0.5254	0.8035	0.7539	0.7869
Hepatitis	0.7412	0.7085	0.6588	0.6810	0.7168	0.5857	0.7244	0.7217	0.7410
Housevotes	0.9505	0.9561	0.9646	0.9633	0.9712	0.9620	0.9506	0.9591	0.9483
Ionosphere	0.8750	0.8536	0.8736	0.8805	0.8205	0.8074	0.9352	0.8828	0.9308
Mammographic	0.7550	0.8107	0.8317	0.8342	0.8252	0.7722	0.8181	0.7453	0.8078
Monk-2	0.7419	0.9509	1.0000	1.0000	0.7756	0.5027	1.0000	0.9995	0.9611
Mushroom	1.0000	1.0000	1.0000	1.0000	0.5000	0.8905	1.0000	1.0000	1.0000
Phoneme	0.8690	0.8490	0.8331	0.8071	0.6837	0.5159	0.8474	0.8268	0.8377
Pima	0.6513	0.6713	0.7047	0.7005	0.7235	0.5044	0.6777	0.7029	0.6837
Saheart	0.5938	0.6067	0.6336	0.6375	0.6772	0.5050	0.6007	0.6250	0.6019
Sonar	0.8575	0.8344	0.7354	0.7796	0.7377	0.5416	0.8735	0.7297	0.8709
Spambase	0.8969	0.8981	0.9214	0.9278	0.8695	0.6334	0.9439	0.9230	0.9339
Spectfheart	0.6217	0.6369	0.6201	0.6008	0.5607	0.5000	0.6666	0.6505	0.7589
Tic-tac-toe	0.9104	0.8956	0.8152	0.9765	0.6510	0.5000	0.9884	0.9731	0.8856
Titanic	0.5227	0.5493	0.6911	0.6754	0.6996	0.5001	0.6826	0.6699	0.6824
Wdbc	0.9534	0.9642	0.9330	0.9452	0.9429	0.8149	0.9695	0.9299	0.9540
Wisconsin	0.9570	0.9640	0.9482	0.9568	0.9509	0.5473	0.9620	0.9606	0.9690

showed a significant difference in step 2 (those marked with bold font), using the p-values obtained in that step as the significance threshold. Count how many of the 100 instances achieve the same results. Table V shows an example table of results for DOB-SCV. A similar table is constructed for each of the other partitioning methods studied. The number on each cell is the number of DOB-SCV partitions where the Wilcoxon signed-ranks test declared a significant difference between the compared classifiers' performance. For example, the 71 in the PART versus 1 NN comparison means that 71 out of 100 independent Wilcoxon signed-ranks test (each one with a different partition) proved significant with a threshold of p-value $= 7.45E-08$. The p-value was obtained from that same cell in Table IV.

4) Average the results of each cell to obtain an aggregated estimation of how close a given partitioning method is to the "true" estimation. In the example of DOB-SCV for 5×2, that average turns out to be 55.684. This is how Table VI is constructed.

To sum up: For each dataset, we averaged the performance (AUC over the test set) of each classifier over the 200 cross-validation experiments, and then performed a Wilcoxon signed-ranks test [26] with a significance level of 0.1 to test whether there existed significant differences between their performances. We considered this Wilcoxon test to be the true comparison between each classifier pair.

To figure out the number of iterations needed for convergence to a stable result, we used the method recommended in [27], which determines the convergence based on reaching a Pearson correlation between accumulated average performances of consecutive instances of cross-validation of 0.9999. More specifically, the method follows these steps in Algorithm 5 (again, using the example of 5×2 with DOB-SCV, the others being analogous).

To achieve more significant results, we repeated this test 10 times for each dataset–classifier pair.

All the experiments were conducted using the KEEL software tool [25].

VI. RESULTS

This section presents a summary of the results obtained by the experiments run following the above framework. Because of space concerns, only a brief summary of the results are included; for a more detailed analysis check http://sci2s.ugr.es/covariate-shift-cross-validation. We first focus on

TABLE IV
WILCOXON SIGNED-RANKS TEST. R+ (R−) DENOTES THE R SCORE OF THE ROW (COLUMN) ALGORITHM. RESULTS ARE PRESENTED AS R+/R−/p-VALUE

	1 NN	3 NN	C45	FURIA	LDA	PART	PDFC	RIPPER	SVM
1 NN	-	119.5/258.5/0.1001	150/228/>0.2	120.5/257.5/0.1029	235/143/>0.2	**375/3/7.45E-08**	**38.5/339.5/9.90E-05**	143.5/234.5/>0.2	**85.0/293.0/1.12E-02**
3 NN	258.5/119.5/0.1001	-	185/193/>0.2	143.5/234.5/>0.2	**261.0/117.0/0.0859**	**377/1/2.98E-08**	**68.5/309.5/2.84E-03**	193.5/184.5/>0.2	120/258/0.1004
C45	228/150/>0.2	193/185/>0.2	-	120.5/257.5/0.1029	242/136/>0.2	**378/0/1.49E-08**	**90.5/287.5/1.68E-02**	180/198/>0.2	146/232/>0.2
FURIA	257.5/120.5/0.1029	234.5/143.5/>0.2	257.5/120.5/0.1029	-	**278.0/100.0/0.0319**	**378/0/1.49E-08**	**108.0/270.0/0.0521**	245.5/132.5/0.1814	185/193/>0.2
LDA	143/235/>0.2	**117.0/261.0/0.0859**	136/242/>0.2	**100.0/278.0/0.0319**	-	**340.0/38.0/**	**77.0/301.0/**	124/254/0.1225	**83.0/295.0/9.61E-03**
PART	**3/375/7.45E-08**	**1/377/2.98E-08**	**0/378/1.49E-08**	**0/378/1.49E-08**	**38.0/340.0/**	-	**1/377/2.98E-08**	**3/375/7.45E-08**	**3/375/7.45E-08**
PDFC	**339.5/38.5/9.90E-05**	**309.5/68.5/2.84E-03**	**287.5/90.5/1.68E-02**	**270.0/108.0/0.0521**	**301.0/77.0/**	**377/1/2.98E-08**	-	**310.5/67.5/**	**273.0/105.0/0.0435**
RIPPER	234.5/143.5/>0.2	184.5/193.5/>0.2	198/180/>0.2	132.5/245.5/0.1814	254/124/0.1225	**375/3/7.45E-08**	**67.5/310.5/**	-	139/239/>0.2
SVM	**293/85/1.12E-02**	258/120/0.1004	232/146/>0.2	193/185/>0.2	**295.0/83.0/9.61E-03**	**375/3/7.45E-08**	**105.0/273.0/0.0435**	239/139/>0.2	-

TABLE V
NUMBER OF TIMES WILCOXON SIGNED-RANKS OBTAINED THE SAME RESULT IN DOB-SCV AS IT DID IN THE "TRUE" RUN

	1 NN	3NN	C45	FURIA	LDA	PART	PDFC	RIPPER	SVM
1 NN	-	-	-	-	-	**71**	**42**	-	**23**
3 NN	-	-	-	-	**96**	**82**	**40**	-	-
C45	-	-	-	-	-	**30**	**60**	-	-
FURIA	-	-	-	-	**72**	**37**	**49**	-	-
LDA	-	**96**	-	**72**	-	**48**	**91**	-	**51**
PART	**71**	**82**	**30**	**37**	**48**	-	**66**	**59**	**56**
PDFC	**42**	**40**	**60**	**49**	**91**	**66**	-	**27**	**58**
RIPPER	-	-	-	-	-	**59**	**27**	-	-
SVM	**23**	-	-	-	**51**	**56**	**58**	-	-

TABLE VI
SINGLE CROSS-VALIDATION EXPERIMENT AND AVERAGE ACCURACY OF THE WILCOXON SIGNED-RANKS TEST

CV scheme	DOB-SCV	DB-SCV	SCV	MS-SCV
2×5	52.687	51.250	31.500	1.250
5×2	55.684	52.737	39.789	0.000
10×1	49.857	51.095	45.333	1.762
Average	52.743	51.694	38.874	1.004

TABLE VII
NUMBER OF CROSS-VALIDATION EXPERIMENTS NEEDED TO CONVERGE TO A STABLE PERFORMANCE ESTIMATION

CV scheme	DOB-SCV	DB-SCV	SCV	MS-SCV
2×5	16.71±12.01	16.82±11.67	18.39±11.88	31.42±13.23
5×2	33.46±24.05	34.86±23.85	37.16±24.41	62.58±26.17
10×1	53.98±32.08	54.70±32.17	59.73±32.14	85.77±27.14

the single-experiment case, in Section VI-A and then present the results corresponding to the number of iterations needed to stabilize in Section VI-B.

A. Single Cross-Validation Example

Table VI shows a summary of the results regarding single-experiment reliability. The data in said table represents the percentage of the time taken by a single cross-validation experiment in comparing two datasets using a Wilcoxon signed-ranks test obtained the right result, which is understood to be the "true" one as defined in Section V. Remember that the significance level is set to the same as the "true" data achieved, and that only comparisons between classifiers that showed a significant difference in performance are considered.

Some interesting conclusions can be extracted by looking at these results.

1) Partition-induced covariate shift can significantly hamper the reliability of running a single experiment. MS-SCV produces a much worse accuracy than all other partitioning strategies.
2) Randomly distributing the examples of a dataset can sometimes induce covariate shift, as can be deduced from SCV having a lower accuracy than both DB-SCV and DOB-SCV.
3) DOB-SCV and DB-SCV obtain similar results, with a slight advantage in favor of DOB-SCV.

B. Number of Iterations Needed to Stabilize

Table VII shows the average number of cross-validation experiments needed to converge to a stable performance estimation, along with the standard deviation.

A number of interesting conclusions can be drawn from this table.

1) It can be seen that covariate shift can potentially have a serious impact in the stability of the results obtained by classifiers, as proven by the difference in iterations needed between MS-SCV and the other methods.
2) The sporadic appearance of covariate shift has an impact in convergence terms, as shown by the fact that both DB-SCV and DOB-SCV converge faster than SCV.
3) 2×5 experiments converge significantly faster than 5×2 and 10×1, and 5×2 also converges significantly faster than 10×1. This is because each instance of the 2×5 method is already comprised of five runs of twofold

Algorithm 5 Convergence Estimation Algorithm

for each method studied m_i in Methods **do**
  for each dataset studied d_j in Datasets **do**
    Estimate the performance of m_i over d_j using DOB-SCV 5×2 cross-validation, saving it in Estim_{ij0}
    Estimate the performance of m_i over d_j using a different instance of DOB-SCV 5×2 cross-validation, saving it in Estim_{ij1}
    $\text{Estim}_{ij1} \leftarrow E_{ij0} \cup \text{Estim}_{ij1}$
    $k \leftarrow 2$
    while PearsonCorrelation(E_{ij0}, E_{ij1})<0.9999 **do**
      $\text{Estim}_{ij0} \leftarrow \text{Estim}_{ij1}$
      $\text{Estim}_{ij1} \leftarrow \text{Estim}_{ij1} \cup \text{Estim}_{ijk}$, where Estim_{ijk} is a new performance estimation obtained with a different DOB-SCV 5×2 instance
      $k \leftarrow k + 1$
    end while
    The convergence for m_i over d_j is defined as k: $\text{Conv}_{ij} \leftarrow k$.
  end for
end for

cross-validation, effectively using more information per iteration than 5×2 and 10×1. An analogous reason explains the difference between 5×2 and 10×1.

VII. Conclusion

We presented an experimental analysis on the impact partition-based covariate shift can have on the reliability of classifier performance through cross-validation.

We studied four different partitioning methods and showed that, when a covariate shift is introduced, single-experiment reliability decreases and the number of iterations required to reach a stable state increases.

We found that cross-validation approaches that try and limit the impact of partition-induced covariate shift are more reliable when running a single experiment, and need a lower number of iterations to stabilize. Among them, we showed that DOB-SCV is slightly more effective than DB-SCV, presenting an example of the type of situation where DOB-SCV can perform better than DB-SCV. We thus recommend cross-validation users to use DOB-SCV as the partitioning method in order to avoid covariate-shift-related problems.

We studied the number of iterations needed to reach a stable performance estimation with the different partitioning strategies, and found that DOB-SCV and DB-SCV outperform the others, which supports the claim that partition-induced covariate shift can hinder the reliability of classifier evaluation and the need for a specifically designed partitioning method to avoid it.

References

[1] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. Int. Joint Conf. Artif. Intell.*, 1995, pp. 1137–1145.

[2] M. Stone, "Asymptotics for and against cross-validation," *Biometrika*, vol. 64, no. 1, pp. 29–35, 1977.

[3] B. Efron and G. Gong, "A leisurely look at the bootstrap, the jackknife, and cross-validation," *Amer. Stat.*, vol. 37, no. 1, pp. 36–48, 1983.

[4] J. Zhang, X. Wang, U. Krüger, and F.-Y. Wang, "Principal curve algorithms for partitioning high-dimensional data spaces," *IEEE Trans. Neural Netw.*, vol. 22, no. 3, pp. 367–380, Mar. 2011.

[5] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2226–2236, Dec. 2011.

[6] H. He, S. Chen, K. Li, and X. Xu, "Incremental learning from stream data," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 1901–1914, Dec. 2011.

[7] J. Q. Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. Cambridge, MA: MIT Press, 2009.

[8] J. G. Moreno-Torres, T. Raeder, R. Aláiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognit.*, vol. 45, no. 1, pp. 521–530, Jan. 2012.

[9] J. R. Cano, F. Herrera, and M. Lozano, "Evolutionary stratified training set selection for extracting classification rules with trade off precision-interpretability," *Data Knowl. Eng.*, vol. 60, pp. 90–108, Jan. 2007.

[10] X. Zeng and T. R. Martinez, "Distribution-balanced stratified cross-validation for accuracy estimation," *J. Experim. Theor. Artif. Intell.*, vol. 12, no. 1, pp. 1–12, 2000.

[11] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *J. Mach. Learn. Res.*, vol. 8, pp. 985–1005, Dec. 2007.

[12] A. Storkey, "When training and test sets are different: Characterizing learning transfer," in *Dataset Shift in Machine Learning*, J. Quiñonero Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, Eds. Cambridge, MA: MIT Press, 2009, pp. 3–28.

[13] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Mateo, CA: Morgan Kaufmann, 2005.

[14] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *J. Stat. Plann. Inference*, vol. 90, no. 2, pp. 227–244, 2000.

[15] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 3, pp. 299–310, Mar. 2005.

[16] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Multiple-Valued Logic Soft Comput.*, vol. 17, nos. 2–3, pp. 255–287, 2010.

[17] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[18] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA: Morgan Kaufmann, 1993.

[19] J. Hühn and E. Hüllermeier, "FURIA: An algorithm for unordered fuzzy rule induction," *Data Mining Knowl. Discovery*, vol. 19, no. 3, pp. 293–319, 2009.

[20] R. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugen.*, vol. 7, no. 2, pp. 179–188, 1936.

[21] E. Frank and I. Witten, "Generating accurate rule sets without global optimization," in *Proc. 15th Int. Conf. Mach. Learn.*, 1998, pp. 144–151.

[22] Y. Chen and J. Wang, "Support vector learning for fuzzy rule-based classification systems," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 6, pp. 716–728, Dec. 2003.

[23] W. Cohen, "Fast effective rule induction," in *Proc. 12th Int. Conf. Mach. Learn.*, 1995, pp. 1–10.

[24] W. C. Cohen, "Fast effective rule induction," in *Proc. 12th Int. Conf. Mach. Learn.*, 1995, pp. 115–123.

[25] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. D. Jesus, S. Ventura, J. M. Garrell, J. Otero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera, "Keel: A software tool to assess evolutionary algorithms for data mining problems," *Soft Comput. Fusion Found., Methodol. Appl.*, vol. 13, no. 3, pp. 307–318, 2009.

[26] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, 2010.

[27] T. Raeder, T. R. Hoens, and N. V. Chawla, "Consequences of variability in classifier performance estimates," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 421–430.

Jose García Moreno-Torres received the M.Sc. degree in computer science from the University of Granada, Granada, Spain, in 2008. He is currently pursuing the Ph.D. degree with the Soft Computing and Intelligent Information Systems Group, Department of Computer Science and Artificial Intelligence, University of Granada, under the supervision of Prof. F. Herrera.

His current research interests include dataset shift, imbalanced classification, and bibliometrics.

Mr. Moreno-Torres was a recipient of an International "La Caixa" Scholarship, and conducted research as a Fellow of the IlliGAL Laboratory, University of Illinois at Urbana-Champaign, Urbana, under the supervision of Prof. D. E. Goldberg.

José A. Sáez received the M.Sc. degree in computer science from the University of Granada, Granada, Spain, in 2009. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Artificial Intelligence, University of Granada.

His current research interests include noisy data in classification, discretization methods, and imbalanced learning.

Francisco Herrera (M'10) received the M.Sc. degree in mathematics and the Ph.D. degree in mathematics from the University of Granada, Granada, Spain, in 1988 and 1991, respectively.

He is currently a Professor with the Department of Computer Science and Artificial Intelligence, University of Granada. He has published over 200 papers in international journals. He is the co-author of the book *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases* (World Scientific, 2001). His current research interests include computing with words and decision making, data mining, bibliometrics, data preparation, instance selection, fuzzy rule-based systems, genetic fuzzy systems, knowledge extraction based on evolutionary algorithms, and memetic algorithms and genetic algorithms.

Prof. Herrera is currently an Editor-in-Chief of the international journal *Progress in Artificial Intelligence* (Springer) and serves as an Area Editor of the *Journal Soft Computing (Area of Evolutionary and Bioinspired Algorithms)* and the *International Journal of Computational Intelligence Systems (Area of Information Systems)*. He is an Associate Editor of several journals, including the IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Information Sciences, Advances in Fuzzy Systems* and the *International Journal of Applied Metaheuristics Computing*. He serves as a member of the editorial boards of many journals, including *Fuzzy Sets and Systems, Applied Intelligence, Knowledge and Information Systems, Information Fusion, Evolutionary Intelligence,* the *International Journal of Hybrid Intelligent Systems,* and *Memetic Computation, Swarm and Evolutionary Computation*. He received the ECCAI Fellowship in 2009, the Spanish National Award on Computer Science ARITMEL to the Spanish Engineer on Computer Science in 2010, and the International Cajastur "Mamdani" Prize for Soft Computing (Fourth Edition) in 2010.

Bibliography

[ARGCCS07] Alaiz-Rodríguez R., Guerrero-Curieses A., and Cid-Sueiro J. (2007) Minimax regret classifier for imprecise class distributions. *Journal of Machine Learning Research* 8: 103–130.

[ARGCCS11] Alaíz-Rodríguez R., Guerrero-Curieses A., and Cid-Sueiro J. (2011) Class and subclass probability re-estimation to adapt a classifier in the presence of concept drift. *Neurocomputing* 74(16): 2614–2623.

[ARJ08] Alaiz-Rodríguez R. and Japkowicz N. (2008) Assessing the impact of changing environments on classifier performance. In *Canadian AI'08: Proceedings of the Canadian Society for computational studies of intelligence, 21st conference on Advances in artificial intelligence*, pp. 13–24. Springer-Verlag, Berlin, Heidelberg.

[BBS09] Bickel S., Brückner M., and Scheffer T. (September 2009) Discriminative learning under covariate shift. *Journal of Machine Learning Research* 10: 2137–2155.

[BFR10] Biggio B., Fumera G., and Roli F. (2010) Multiple classifier systems for robust classifier design in adversarial environments. *International Journal of Machine Learning and Cybernetics* 1: 27–41.

[BPM04] Batista G. E. A. P. A., Prati R. C., and Monard M. C. (2004) A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations Newsletter* 6(1): 20–29.

[CB04] Crook J. and Banasik J. (2004) Does reject inference really improve the performance of application scoring models? *Journal of Banking & Finance* 28(4): 857–874.

[CBHK02] Chawla N. V., Bowyer K. W., Hall L. O., and Kegelmeyer W. P. (2002) SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16: 321–357.

[CC09] Cieslak D. A. and Chawla N. V. (2009) A Framework for Monitoring Classifiers' Performance: When and Why Failure Occurs? *Knowledge and Information Systems* 18(1): 83–108.

[CK05] Chawla N. and Karakoulas G. (2005) Learning from labeled and unlabeled data: An empirical study across techniques and domains. *Journal of Artificial Intelligence Research* 23(1): 331–366.

[DDM+04] Dalvi N., Domingos P., Mausam, Sanghai S., and Verma D. (2004) Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on*

Knowledge discovery and data mining, KDD '04, pp. 99–108. ACM, New York, NY, USA.

[Die00] Dietterich T. G. (2000) Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, pp. 1–15. Springer-Verlag, London, UK, UK.

[DT10] Denil M. and Trappenberg T. P. (2010) Overlap versus Imbalance. In *Canadian Conference on AI*, pp. 220–231.

[Elk01] Elkan C. (2001) The Foundations of Cost-Sensitive Learning. In *IJCAI*, pp. 973–978.

[FF05] Fawcett T. and Flach P. A. (2005) A response to Webb and Ting's 'On the application of ROC analysis to predict classification performance under varying class distributions'. *Machine Learning* 58(1): 33–38.

[FGdJH08] Fernández A., García S., del Jesus M. J., and Herrera F. (2008) A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems* 159(18): 2378–2398.

[GH09] García S. and Herrera F. (2009) Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. *Evolutionary Computation* 17(3): 275–306.

[GMS08] García V., Mollineda R. A., and Sánchez J. S. (2008) On the k-NN performance in a challenging scenario of imbalance and overlapping. *Pattern Analysis & Applications* 11(3-4): 269–280.

[GSH+09] Gretton A., Smola A., Huang J., Schmittfull M., Borgwardt K., and Schölkopf B. (2009) Covariate Shift by Kernel Mean Matching. In Quiñonero Candela J., Sugiyama M., Schwaighofer A., and Lawrence N. D. (Eds.) *Dataset Shift in Machine Learning*, pp. 131–160. The MIT Press.

[Han98] Hand D. (1998) Reject inference in credit operations. *Credit risk modeling: Design and application* pp. 181–190.

[Hec79] Heckman J. (1979) Sample selection bias as a specification error. *Econometrica: Journal of the econometric society* pp. 153–161.

[HG09] He H. and Garcia E. A. (September 2009) Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering* 21(9): 1263–1284.

[KHS09] Kanamori T., Hido S., and Sugiyama M. (December 2009) A least-squares Approach to Direct Importance Estimation. *Journal of Machine Learning Research* 10: 1391–1445.

[Koh95a] Kohavi R. (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 1137–1143. Morgan Kaufmann.

[Koh95b] Kohavi R. (1995) A study of Cross-Validation and bootstrap for accuracy estimation and model selection. In *International Joint Conferences on Artificial Intelligence*, pp. 1137–1145.

[LL10] Laskov P. and Lippmann R. (2010) Machine learning in adversarial environments. *Machine Learning* 81: 115–119.

[MTH10] Moreno-Torres J. G. and Herrera F. (2010) A preliminary study on overlapping and data fracture in imbalanced domains by means of genetic programming-based feature extraction. In *ISDA*, pp. 501–506. IEEE.

[MTLGB13] Moreno-Torres J. G., Llorà X., Goldberg D. E., and Bhargava R. (2013) Repairing Fractures between Data using Genetic Programming-based Feature Extraction: A Case Study in Cancer Diagnosis. *Information Sciences* 222: 805–823.

[MTSH12] Moreno-Torres J. G., Sáez J. A., and Herrera F. (2012) Study on the impact of partition-induced dataset shift on k-fold cross-validation. *IEEE Transactions on Neural Networks and Learning Systems* 23(8): 1304–1312.

[QnCSSL09] Quiñonero Candela J., Sugiyama M., Schwaighofer A., and Lawrence N. D. (2009) *Dataset Shift in Machine Learning*. The MIT Press.

[Shi00] Shimodaira H. (2000) Improving predictive inference under Covariate Shift by Weighting the Log-likelihood Function. *Journal of Statistical Planning and Inference* 90(2): 227–244.

[SKM07] Sugiyama M., Krauledat M., and Müller K.-R. (2007) Covariate Shift Adaptation by Importance Weighted Cross Validation. *Journal of Machine Learning Research* 8: 985–1005.

[SKWW07] Sun Y., Kamel M. S., Wong A. K., and Wang Y. (2007) Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition* 40(12): 3358 – 3378.

[Sto77] Stone M. (1977) Asymptotics For and Against Cross-Validation. *Biometrika* 64(1): 29–35.

[SWK09] Sun Y. M., Wong A. K. C., and Kamel M. S. (2009) Classification of Imbalanced Data: A Review. *International Journal of Pattern Recognition and Artificial Intelligence* 23(4): 687–719.

[WK96] Widmer G. and Kubat M. (1996) Learning in the presence of concept drift and hidden contexts. *Machine Learning* 23: 69–101.

[WZF+03] Wang K., Zhou S., Fu C. A., Yu J. X., Jeffrey F., and Yu X. (2003) Mining changes of classification by correspondence tracing. In *Proceedings of the 2003 SIAM International Conference on Data Mining (SDM 2003)*, pp. 95–106.

[XQ07] Xie J. and Qiu Z. (2007) The effect of imbalanced data sets on LDA: A theoretical and empirical analysis. *Pattern Recognition* 40(2): 557 – 562.

[YWZ08] Yang Y., Wu X., and Zhu X. (2008) Conceptual equivalence for contrast mining in classification learning. *Data & Knowledge Engineering* 67(3): 413–429.

[Zad04] Zadrozny B. (2004) Learning and evaluating classifiers under sample selection bias. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 114. ACM, New York, NY, USA.