Information Sciences 579 (2021) 388–400

Contents lists available at ScienceDirect

Information Sciences

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

Multiple instance classification: Bag noise filtering for negative instance noise cleaning

Julián Luengo [a,*], Dánel Sánchez-Tarragó [b], Ronaldo C. Prati [c], Francisco Herrera [a]

[a] Dept. of Computer Science and Artificial Intelligence, University of Granada, Granada 18071, Spain
[b] Department of Computer Science, Central University "Marta Abreu" of Las Villas, Santa Clara, Cuba
[c] Center of Mathematics, Computer Science and Cognition (CMCC) Federal University of ABC (UFABC), Santo André, São Paulo, Brazil

ARTICLE INFO

Article history:
Received 24 March 2019
Received in revised form 23 July 2021
Accepted 25 July 2021
Available online 27 July 2021

Keywords:
Multiple instance classification
Data preprocessing
Noisy data
Instance noise
Noise filtering

ABSTRACT

Data in the real world is far from being perfect. The appearance of noise is a common issue that arises from the limitations of data acquisition mechanisms and human knowledge. In classification, label noise will hinder the performance of almost all classifiers, inducing a bias in the built model. While label noise has recently attracted researchers' attention in standard classification, it has only recently begun to be studied in multiple instance classification. In this work, we propose the usage of filtering algorithms for multiple instance classification that are able to reduce the impact of negative instances within the bags. In order to do so, we decompose the bags to form a standard classification problem that can be efficiently treated by a specialized noise filter. Such a decomposition is tackled in different ways, with the aim of exploiting the knowledge offered by the examples from opposite bags. The bags are then rebuilt, without the identified noise instances. In our experiments, we show that by applying our approach we can diminish the impact of noise and even obtain better results at 0% noise level for several classifiers. Our approach sets out a promising approach to dealing with noise in the bags of multiple instance datasets and further improve the classification rate of the built models.

1. Introduction

In the last decade, real-life problems have led to new and extended classification paradigms to be established beyond standard classification [22]. In machine learning, multiple instance learning (MIL) [23] is a generalization of the classic attribute–value paradigm, where each learning object is composed by many vectors. The generalization of MIL affects several research areas in Data Science, which have been well studied in standard problems but need a whole new methodological approach under MIL [5]. One prominent topic of MIL is multiple instance classification (MIC) [2]. In standard classification, the instances of a dataset are described by a vector of attribute values with an associated label. On the other hand, in MIC, the samples are *bags* of instances [1]. Each bag comprises a number of instances with only their input features. This number may vary from bag to bag. All the instances within a bag correspond to alternative descriptions of the concept represented by the bag, just like several atomic structures for the same molecule. The instance in the bag does not have an outcome associated

* Corresponding author.
E-mail addresses: julianlm@decsai.ugr.es (J. Luengo), danels@uclv.edu.cu (D. Sánchez-Tarragó), ronaldo.prati@ufabc.edu.br (R.C. Prati), herrera@decsai.ugr.es (F. Herrera).

https://doi.org/10.1016/j.ins.2021.07.076
0020-0255/© 2021 The Authors. Published by Elsevier Inc.

to it, it only belongs to its respective bags. MIC applications range from image classification in medical domains [20,39], speech recognition [34] or biometric recognition [11].

The complex nature of MIC datasets have led to the creation of custom algorithms for diverse tasks, from image tracking [14] to video classification [26], or even the adaptation of preprocessing techniques from standard paradigms, such as feature selection [5]. While a large set of classifiers have been developed to tackle MIC, there are still open-ended problems that are well-known in standard classification that require attention in MIC. Among them, learning from noisy data is an important topic in machine learning that remains open in MIC. Noisy data has attracted a considerable amount of attention and the research community have developed numerous methods to deal with it [13,48]. Thus, there is a large range of literature regarding class noise [31] and handling the noise in standard classification problems. These approaches include the development of robust algorithms for noise and data preprocessing methods to clean the noise in the data. Although some authors have analyzed which current MIL classifiers act better against noise [26], or have tried to develop novel noise-tolerant MIL classifiers [7], the solution has not been tackled yet, constituting an open-ended challenge.

We face the problem of noisy instances within the bags in MIC problems, based on the decomposition of the bags in a single-instance problem, where a standard well-known noise filter is applied. In [28], we explored the simple MIL-Filtering bag filter, comprised of the Single Transforming Filter with IPF [24] as a base filter, which showed good potential.

In this paper, we fully tackle this promising preprocessing approach, proposing a whole new methodology with two novel bag filtering decomposition approaches: the Nearest Neighbor Transforming Filter and the Nearest Neighbor Multi-Wrapper Filter. We also use three standard noise filters as base filters: IPF, EF [4] and RNG [36], thus creating and proposing *six new bag filters* to be thoroughly compared and analyzed, including the filter of [28] as well.

In order to validate the proposed filters, a wide experimental setting has been carried out. We introduce several noisy instances in the bags at different levels, ranging from 5% to 20%, over twenty-eight datasets. The performance results of four different MIL classifiers are compared to evalute the potential of the proposed bag filters, along with performing no filtering at all. In general, the results show that in most cases, using a MIL noise filter is beneficial. In particular, some of the proposed filters in this paper are more suited to certain MIL classifiers and noise levels. The benefits are significant from a low noise level, such as 10% noise onwards, which enables our method to be safely used in many noisy scenarios.

The rest of this paper is organized as follows. First, Section 2 introduces the MIL paradigm along with classification in MIL. Section 3 presents an introduction to classification with noisy data, both in standard and MIL problems. Section 4 introduces the proposed filtering methods. Section 5 presents the experimental framework, and Section 6 analyzes the performance results obtained. Section 7 studies the behavior of the proposed methods in terms of the number of filtered instances. Finally, Section 8 enumerates the concluding remarks drawn from this work.

2. Classification in multiple instance learning

The multiple instance learning (MIL) paradigm was introduced in the seminal work of [9]. It arose in the context of learning tasks where there are multiple descriptions for each example of the concept that needs to be learned. For example, it can be used when we have multiple viewpoints of the same object, when we have one description for each part of an object composed of multiple parts, or when we have observations taken at different points in time of an object that is changing. In each of these cases, several descriptions can be associated with the same object (i.e., the example).

In MIL terminology, we say that an example is a bag of instances where each instance corresponds to one of the descriptions associated with the example. In order to train a learning algorithm, we use a training set of bags where a supervisor agent has previously assigned each bag to a class label. While the class label of each training bag is available for the learner, the class label of each instance remains unknown. Fig. 1 shows the difference between training examples from MIL and those from the traditional (single-instance) learning.

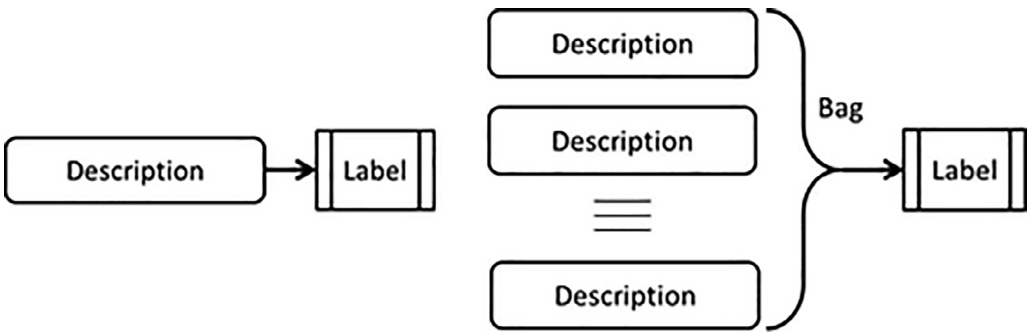

Fig. 1. Each example in Single Instance Learning (left) has only one description and a corresponding class label. On the other hand, each example in Multiple Instance earning (right) has many descriptions and only one class label associated to all of them.

For the purpose of this paper, we state the following formal description of MIL. In order to carry out the learning task, we are given a training set of bags. This training set consists of a set of N pairs of data $\{(X_1, y_1), \ldots, (X_N, y_N)\}$ where X_i is the bag in the i-th pair and yi is the class label associated to X_i. Each bag X_i is a set that contains an specific number M_i of instances, i.e., $X_i = \{x_1, \ldots, x_{Mi}\}$. Each instance $x_j \in X_i$ is a vector of dimension S where the k-th component of the vector corresponds to the value of the descriptive attribute A_k for the instance x_j, i.e., $\forall k \in \{1, \ldots, S\} : x_{jk} \in A_k$. We focus this study on two-class MIL problems, which have received great attention in the specialized literature, therefore $y_i \in \{0, 1\}$ where 0 is the label that represents the negative class and 1 represents the positive class.

Although multiple instance representation gives us a very natural and flexible way to represent complex learning objects, solving MIC problems is complicated, since the relationship between instances and bags differs depending on the problem itself. In some cases, only one instance is needed to label a bag. This assumption is referred as the *standard* and implies that the learning process needs to consider that the bag can have both instances representing the concept to learn and others that do not.

Several authors have proposed a varied kind of classifiers that have has been adapted to deal with MIL problems in classification. Following Amores taxonomy [2], the classifiers can be categorized as:

- Instance space paradigm: the classifier creates the discriminant function in the instance space.
- Bag space paradigm [6]: the classifier works with the whole bag by means of similarity functions.
- Embedded space paradigm [25]: the classifier transform the original space to a new embedded space, where the bags are represented as single attribute vectors.

In this study we will select a set of representative classifiers from different categories from the aforementioned taxonomy, summarized as follows:

- **SimpleMI** [43] transforms each bag to a onedimensional vector. C4.5 [33] is applied to the transformed dataset.
- **MITI** [3] is a native decision tree method for MIL problems based on ID3 that expands nodes in a best-first criterion.
- **MILR** [10] is the logistic regression adaptation to MIL.
- **CitationKNN** [43] extends the notion of neighborhood by considering not only the nearest neighbor bags, but also the bags that consider the actual bag as neighbor itself. This extension is coupled with the usage of Hausdorff distance as well.
- **MIboost** is an extension of AdaBoost that considers the geometric mean of posterior of instances inside a bag (arithmetic mean of log-posterior) and the expectation for a bag is taken inside the loss function.

The implementation of these methods can be found in the WEKA software framework [10].

While the standard assumption only requires a single instance within the bag to label the latter, it is desirable to examine whether the other instance are prejudicial or not for the MIL classifiers. In this study we have explored the possibility of cleaning the bags to improve the performance of MIL classifiers, particularly when the bags are contaminated with noisy examples that hinder the bag description.

3. Noisy data in MIL classification

This section offers a general overview of noisy data for the classification task in Section 3.1. It is followed by a general introduction to noise filters in Section 3.2.

3.1. Introduction to noisy data

Real-world data is far from being perfect and accurate [21]. As indicated in [18], the presence of noise in data can affect the complexity of classification problems. All these imperfections may harm the interpretation of data, the design, size, building time, interpretability and accuracy of models, as well as making decisions [47].

To diminish the effects of noise, it is crucial to identify the components that can be affected by its presence. In standard classification, class labels and attribute values are pieces of information that can be affected, thus two main types of noise are distinguished in the literature [48]: class noise (or label noise) and attribute noise. *Class noise* occurs when an instance belongs to the incorrect class either as a result of contradictory examples [21] or misclassifications [48]. It can be attributed to several causes, including subjectivity during the labeling process, data entry errors, or inadequacy of the information used to label each instance. *Attribute noise* refers to corruptions in the values of one or more attributes in a dataset. Examples of attribute noise include erroneous attribute values, missing or unknown attribute values and "do not care" values. As the identification and reparation of noise has been demonstrated to be beneficial to the classification task [29], tackling the problem of noise in non-standard classification problems is drawing more and more attention in the specialized literature [32].

If we focus on MIL, we can think of such a paradigm as a way of modeling the data based on a two-level description. At a basic level, there are instances describing the attribute values. At a higher level, there are bags, each with its own class label. Noise can be present at both levels: at the bag level, affecting the class labels of bags, and at the instance level, damaging the attribute values.

Like label noise in traditional single-instance learning, noise affecting bag labels is caused fundamentally by the erroneous labeling of training examples. In the binary case, this means that a bag that does not have any positive instances is incorrectly labeled as positive, or that a bag, which has at least one positive instance, is incorrectly labeled as negative. However, we can expect label noise to have a greater impact in MIL than it would in single-instance learning. This is due to the fact that in MIL each bag that is incorrectly labeled has a misleading effect, not on a single attribute vector, but rather on each instance of that bag, potentially resulting in (depending on the learning algorithm) a multiplied misleading effect on the learner.

It is also possible to analyze noise at the instance level in MIL. MIL can be seen as a noise dealing technique for one-tier noise (noise within only the positive class), where negative instances inside a positive bag can be considered as a kind of "noisy instances" of the positive concept [27]. A common approach is to estimate the diversity density estimator [30], where the objective is to build a model which takes the evidence required to classify positive instances from the union of positive bags into account, filtering out its intersection with negative bags. The frequent existence of several redundant descriptions for each bag suggests that MIL is more robust than single-instance learning when exposed to the harmful effect of attribute noise. However, this theoretical analysis needs to be supported by empirical studies.

The instances appearing in bags labeled negatively cannot be used to create the bag's concept and can actually make the learning process more difficult. This means that these instances, therefore, function as if they were noise, suggesting that a third type of noise also exists in MIL involving both levels at the same time –instance level and bag level– and affecting the presence of instances inside the bags. We call this new type of noise *instance noise* and we define it in the following way: an instance inside a bag can be considered noise if the information conveyed by the instance hinders the learning task.

To the best of our knowledge, we have not yet found any proposals in the literature that deal with noise in MIL. We could use the popular techniques that deal with noise in standard noise as a point of reference. In particular, among data level techniques, *noise filters* and *data correcting methods* are the two popular options used as a preprocessing step before training a (possibly noise-sensitive) learner. Noise Filters [4,24,42] identify and remove noisy instances from the training data. Data correcting methods [40] aim to correct or repair noisy instances prior to building a learner by relabeling the wrong class labels. Several studies [48,46] claim that complete or partial noise correction in training data, with test data still containing noise, improves performance results, as compared to no preprocessing.

Noise filters are more popular than data correcting methods [13] and the current state-of-the-art will be further elaborated in the following section. However, we are interested in making use of the advantages of well-established noise filtering methods to clean the noisy instances within the bags. Since we focus on the study of instance noise in this paper, we will briefly present the selected preprocessing filtering algorithms in the next section as it is a key part of our proposal.

3.2. Noise filters

Noise filters are preprocessing mechanisms designed to detect and eliminate instances with class noise in the training set [4,24]. The advantage of separating noise detection and learning is that noisy instances do not influence the classifier design [16]. Elimination of instances with class noise has been shown to be advantageous [15], but the elimination of instances with attribute noise seems to be counterproductive [48], since they still contain valuable information in other attributes which can help to build the classifier.

Noise filters follow three main paradigms:

1. Some of them are based on the computation of different *measures* on the data. For instance, Gamberger et al. [16] propose that eliminating noisy examples reduces the "Complexity of the Least Complex Correct Hypothesis" value of the training set.
2. *Similarity-based* approaches rely on a local neighborhood of the instance in order to identify it as noisy or clean. The well-known ENN [45] and its extension ENNTh [41] are examples of this approach.
3. *Ensemble-based* methods collect predictions from different base classifiers in order to estimate the mislabeled examples. Some of the most popular algorithms are EF [4], CVCF [42], and IPF [24].

Research on noise filters is an active topic nowadays. For example, INFFC [35] was proposed in an attempt to integrate the classical paradigm of ensemble-based filtering with both an iterative scheme and a metric-based approach. There are also recent contributions that highlight classification accuracy enhancement after class noise treatment [12]. Another recent study is that of Sluban et al. [38]. They propose creating a ranking of noisy instances according to the predictions provided by several different noise detection algorithms. Improvements in the behavior of existent filters is studied in [17], where the usage of decomposition strategies is shown to be beneficial in improving the behavior of any noise filter.

4. Bag noise filtering for *instance noise*

Since *instance noise* in MIL has not been tackled yet, we present two noise filter algorithms that are based upon the same principle of moving from MIL into a single-instance representation. Once we have a single instance representation, a classic noise filter is applied and then the filtered data is transformed back into the MIL representation.

The transformation from MIL to a single instance problem is performed by augmenting each instance's attribute vector with the addition of the bag's class label. This is a common transformation (and initialization) method in instance based classification algorithms [23].

By following this principle, we propose and describe two novel MIL noise filters: Single Transforming Filter (Section 4.1) and Nearest Neighbor Transforming Filter (Section 4.2). Section 4.3 briefly describes an extension of the latter where several filters cooperate to filter noise.

4.1. Single Transforming Filter

In Single Transforming Filter, all of the training bags are converted to the single-instance representation in a single step. As a result, the information describing which instances belong to which bags (membership information) is lost. The filter only knows about the class label assigned to each instance, but it doesn't know if two instances with the same class label belong to two different bags or to the same bag. The steps of the Single Transforming Filter are shown in Algorithm 1. By means of the loop of Step 3 we complete the transformation process and set the single-instance representation of the data in D. We apply the filter (in Step 11) and remove noisy instances (in Step 12) repeatedly until the detected level of noise (checked in Step 14) falls below a given threshold. Note that when we return to Step 11 to repeat the entire process, D has been previously updated in Step 12. This filter is similar to the *MIL-filtering* technique introduced in [28] if IPF is used as a base filter.

Algorithm 1. Single Transforming Filter

1: **function** STF (T – training data, δ – threshold of filtered instances to stop, F – a standard classification noise filter
2: $D \leftarrow \{\})$
3: **for** all bags $B_i \in T$ **do**
4: Let y_i be the label of B_i
5: **for** all instances $b_j \in B_i$ **do**
6: Create instance $x \leftarrow \{b_j, y_i\}$ as a standard classification instance
7: $D \leftarrow D \cup x$
8: **end for**
9: **end for**
10: **repeat**
11: $N = F(D)$, N being the set of noisy instances identified by F
12: $D \leftarrow D \setminus N$
13: $T \leftarrow T \setminus N$
14: **until** the percent of noisy instances identified in Step (11) $< \delta$
15: **end function**

4.2. Nearest Neighbor Transforming Filter

To test whether the membership information can improve the filtering process of the noisy instances inside the bags or not, we propose the Nearest Neighbor Transforming Filter. In this case, the transformation of the training bags is not made all at once, but instead over multiple steps. In each step, only two bags, one from each class, are transformed and then filtered. The transformation and filtering process is repeated for all the training bags. For each training bag the vote of its nearest neighbors is used to determine the noisy instances of that bag. In this way, the filter is applied to the whole training set, taking into account the membership information.

In Algorithm 2 we show an algorithmic description of the Nearest Neighbor Transforming Filter. For each instance $x_i \in X$we use a variable C_i to count the positive votes of the neighbors of X. A positive vote means that x_i has been identified as noisy by a neighbor. In Step (7) we set the value of each C_i to zero. In Step (3) we need to find the nearest neighbors of X among the bags of the opposite class. In order to do this, we need to use a distance function defined for bags. Specifically, we use the Average Hausdorff Distance [44] as the bag-wise distance function and the Euclidean distance to measure the distance between instances. In Step (5) we perform the transformation process of X and its neighbor Z and set the resulting single-instance representation in D. After applying the filter in Step (9), we increase the counter C_i to each instance x_i that has been deemed as noisy by F (Step (10)). In Step (14) we consider the removal of each instance x_i based on the votes given by the k nearest neighbors of X. Different approaches can be used to make this decision. For example, the *unanimity* policy needs all neighbors to give a positive vote; the *minimum* policy is when the vote of a single neighbor is enough to remove the instance, and the *majority* policy requires more positive votes from the neighbors than negative votes. Each policy defines a threshold γ above which the instance is removed. In the experimental section of this paper we report results using the maximum policy. Each of the steps is repeated until the detected level of noise (checked in Step (14)) falls below a given threshold δ.

The noise filter F to be used in these algorithms can be any regular (instance-level) noise filter, such as those introduced in Section 3.2.

Algorithm 2. Nearest Neighbor Transforming Filter

1: **function** NNTF (T - training data, k – number of nearest neighbor, γ – threshold to remove instances, δ – threshold of filtered instances to stop, F – standard noise filter)
2: **for** each bag X in T: **do**
3: Let N_k be the k nearest bags to X with the opposite class.
4: **for** each bag Z in N_k: **do**
5: Create a single-instance data set D from the instances in X and Z, assigning the class label of each bag to the instances of that bag.
6: **for** each instance x_i in X
7: $C_i \leftarrow 0$
8: **end for**
9: Apply a regular noise filter F to D.
10: $C_i \leftarrow C_i + 1$ to each instance $x_i \in X$ identified as noisy by F.
11: **end for**
12: **for** each instance x_i in X **do**
13: **if** $C_i \geqslant \gamma$ **then**
14: $X \leftarrow X \setminus x_i$
15: **end if**
16: **end for**
17: **end for**
18: **if** the percent of noisy instances identified in Step (9) $\geqslant \delta$ **then**
19: Goto Step 2.
20: **end if**
21: **end function**

4.3. Nearest Neighbor Multi-Wrapper filter

The operation of the Nearest Neighbor Mult-Wrapper filter (MW) is similar to NNTF. Instead of applying one filter to detect noisy instances, several filters are used and then a voting scheme is carried out to decide whether an instance is noisy or not. Algorithm 3 shows the algorithmic description of MW, where the set of filters $\mathscr{F}$ can be any odd combination of standard noise filters.

Algorithm 3. Nearest Neighbor Multi-Wrapper Filter

1: **function** MW (T – training data, k – number of nearest neighbor, γ – threshold to remove instances, δ – threshold of filtered instances to stop, $\mathscr{F}$ – set of standard noise classifiers)
2: **for** each bag X in T: **do**
3: Let N_k be the k nearest bags to X with the opposite class.
4: **for** each bag Z in N_k: **do**
5: Create a single-instance data set D from the instances in X and Z, assigning the class label of each bag to the instances of that bag.
6: **for** each instance x_i in X **do**
7: $C_i \leftarrow 0$
8: $V_i \leftarrow 0$
9: **end for**
10: **for** each noise filter $F \in \mathscr{F}$
11: Apply a regular noise filter F to D.
12: $C_i \leftarrow C_i + 1$ to each instance $x_i \in X$ identified as noisy by F
13: **if** $C_i \geqslant \gamma$
14: $V_i \leftarrow V_i + 1$
15: **end if**
16: **end for**
17: **end for**

(*continued on next page*)

```
18:       for each instance x_i in X do
19:          if V_i ⩾ α then
20:             X ← X \ x_i
21:          end if
22:       end for
23:    end for
24:    if the percent of noisy instances identified in Step (9) ⩾ δ
25:       Goto Step 2.
26:    end if
27: end function
```

By using MW we want to examine whether the knowledge of diverse noisy instances from several filters is compatible. In this study, we will use the set of EF [4], IPF [24] and RNG [36] filters, which are all different in nature, as indicated in Section 3.2.

5. Experimental framework

In this section we present the details of the experimental framework used to validate our proposal. In Section 5.1 we present the datasets used in the experimentation. Section 5.2 details the parameters utilized to analyze the effect of filtering. Finally, Section 5.3 describes the statistical tests that support the conclusions made in the analysis.

5.1. Datasets

In order to check the validity of our proposal, we have used twenty-eight multiple instance datasets with bags belonging to two different classes. These datasets belong to different real-world domains: bioinformatics, text recognition and image classification. They are described in Table 1 and are aimed at offering a representative test-bed for our proposal. The number of bags with their instances are indicated. We also show the number of positive and negative bags contained in each dataset.

To validate the behavior of our proposal in different noisy scenarios, we have introduced several levels of noise in the aforementioned datasets. From each bag of each training partition, a percentage of instances were randomly taken and replaced by instances from bags labeled with the opposite class label. Test partitions in datasets remained identical to

Table 1
Datasets.

Name	attrs	-inst	+inst	inst	-bags	+bags	bags
Musk1	166	269	207	476	45	47	92
Musk2	166	6598	5581	12179	62	39	101
Atoms	10	545	1073	1618	63	125	188
Bonds	16	1040	2955	3995	63	125	188
Chains	24	1233	4116	5349	63	125	188
Elephant	230	629	762	1391	100	100	200
Fox	230	673	647	1320	100	100	200
Tiger	230	676	544	1220	100	100	200
TREC1	320	1644	1580	3224	200	200	400
TREC2	303	1629	1715	3344	200	200	400
TREC3	324	1620	1626	3246	200	200	400
TREC4	306	1637	1754	3391	200	200	400
TREC7	300	1621	1746	3367	200	200	400
TREC9	299	1616	1684	3300	200	200	400
TREC10	303	1635	1818	3453	200	200	400
WIR7	303	1713	1710	3423	58	55	113
WIR8	303	1713	1710	3423	58	55	113
WIR9	301	1713	1710	3423	58	55	113
Corel01vs02	9	354	484	838	100	100	200
Corel01vs03	9	310	484	794	100	100	200
Corel01vs04	9	759	484	1243	100	100	200
Corel01vs05	9	200	484	684	100	100	200
Corel02vs03	9	310	354	664	100	100	200
Corel02vs04	9	759	354	1113	100	100	200
Corel02vs05	9	200	354	554	100	100	200
Corel03vs04	9	759	310	1069	100	100	200
Corel03vs05	9	200	310	510	100	100	200
Corel04vs05	9	200	759	959	100	100	200

the originals, i. e., without introducing noise. The procedure to introduce noise is as follows: for each bag in the training partition, a $x\%$ number of instances were substituted by instances from bags of the opposite class chosen at random. In this way, we ensure that each bag contains a level of noise of $x\%$ each. We have considered $5\%,\ 10\%,\ 15\%$ and 20% noise levels.

The performance of the classifiers is evaluated by means of the accuracy. We have applied a 5-fold cross validation to obtain the accuracy of the methods.

5.2. Parameters

In this section we present the parameters, both filters and classifiers, used by the methods.

Table 2 shows the parameters used in our experimentation for the classification methods. They have been chosen from the recommended values indicated by their respective authors.

Table 3 contains the parameters utilized by the base filters (EF, IPF and RNG). Since the proposed MIL filters can be considered to be wrappers around these base filters, we have written the acronyms of Simple Transforming Filter (SW), Nearest Neighbor Transforming Filter (NN) and Nearest Neighbor Multi-Wrapper (MW) using 'W' for *wrapper*. Thus, the combinations of filters that we will consider in our experiments are those indicated in Table 4.

5.3. Statistical tests

To properly analyze these performance results, Friedman's Statistical Test is used [8]. Friedman test [37] is a non-parametric test equivalent of the repeated-measures ANOVA. Under the null hypothesis, it states that all the algorithms are equivalent, so a rejection of this hypothesis implies the existence of differences among the performances of all the algorithms studied.

After this, a post hoc test could be used in order to find whether the control or proposed algorithm presents statistical differences with the compared methods. In this paper we will consider *Holm's test*, as it is able to adjust the *p*-values to the considered number of algorithms.

The usage of non-parametrical statistical tests is recommended in these types of comparisons since the conditions to apply parametrical tests are not usually fulfilled [19].

6. Analysis on the effect of filtering in classification accuracy

In this section, we study how the noise affects the performance of the MIL classifiers presented in Section 2. The results of filtering with the proposed techniques will be also analyzed, focusing on whether the application of the novel filters help the filters to overcome the negative effects induced by noise. Table 5 shows the accuracy results for the separate classifiers after applying the different filters and the no-filter strategy.

In the case of 0% noise level, we aim to check whether the application of the proposed noise filter is safe, that is, the accuracy achieved after applying the filter is at least comparable to not filtering (NF). The results indicate that NF is the best option for SimpleMI and CitationKNN. However, other filters are beneficial even when no artificial noise is added, showing better results for MITI, MILR and MIBoost. Please note that the later are more sophisticated classifiers, thus suggesting that they are able to benefit more from the filtering of the original datasets. This may also suggest that the original bags contain significant amounts of negative examples that hinder the bag description label, prior to adding any extra amount of artificial noise.

At a 5% noise level, the NF option is only suitable for SimpleMI. Since the amount of noise is still low, the bag codification is only slightly altered and allows C4.5 to cope with these changes. Please remember that C4.5 embedded in SimpleMI is done so by applying pruning, thus making it robust to small amounts of noise. For the other classifiers, the application of a filter is recommended to overcome the negative effects of noise. Except for MITI, using IPF as a base filter is the best technique in terms of accuracy.

At a 10% noise level onward, NF is never the best option in terms of average accuracy. Thus, from medium to high intra-bag noise levels, the application of noise filters is necessary to alleviate the hindering effects of negative examples in the bags. Even at a 10% noise level, where the probability of maintaining positive examples in the bags is remarkable, introducing new negative examples severely spoils the relationship with the bag label. Thus, applying noise filters is recommended when the description of the bags is not accurate enough.

Table 2
Parameters used by the MIL classifiers.

Method	Parameters
SimpleMI	confidence: 0.25, pruning: true
MITI	split: Gini, n: 0.5
MIBoost	classifier: C4.5, iterations: 10
CitationKNN	references: 1, citers: 1, rank Hausdorff distance: 1

Table 3
Parameters used by the filters.

Method	Parameters
EF	Voting scheme: consensus, Γ: 4, K_{EF}: 1, distance: HVDM, prune: true, confidence: 0.1, itemsPerLeaf: 2
IPF	Voting scheme: majority, Γ: 5, K_{IPF}: 3, P_{IPF}: 1%
RNG	graph order: 1^{st} order, selection: edition, distance: Euclidean
SW filter	γ: 3, δ: 1%
NN filter	k: 5, γ: 3, δ: 1%
MW filter	k: 5, γ: 3, δ: 1%, filter voting: simple ($\alpha = 1$)

Table 4
Combinations of base filters and proposed MIL filtering techniques used in the experiments.

	EF	IPF	RNG
Simple Transforming Filter	SW-EF	SW-IPF	SW-RNG
Nearest Neighbor Transforming Filter	NN-EF	NN-IPF	NN-RNG
Nearest Neighbor Multi Wrapper Filter		MW	

To better support our analysis, results from the application of Friedman's statistical tests are available in Table 6. In this table, the best ranked algorithm is denoted as **Control**, while the rejection of the null-hypothesis of equality is codified by using '+', '*' and '=' symbols. A '+' indicates that the null-hypothesis is rejected with a $p - value \leqslant 0.05$, a '*' denotes that the null-hypothesis is rejected with a $p - value \leqslant 0.1$ and '=' indicates that the null-hypothesis is not rejected with a desirable α level.

Since the statistical tests are performed over the accuracy results, some conclusions are similar to those we have drawn from the averaged accuracy scores: the application of NF is allegedly the best choice for CitationKNN and SimpleMI, but with a similar performance to other filters as SW-IPF and SW-EF. Therefore, we may consider the last two to be techniques that are safe for use with the aforementioned classifiers even when the amount of noise is unknown.

By focusing on the most frequent control algorithm, SW-IPF emerges as the preferred filter for MIBoost, SimpleMI (from 5% noise onward). Other classifiers such as MILR, MITI and Citation KNN show SW-IPF as the best choice in some cases. However, for the latter, even when SW-IPF is not the best algorithm, it is not significantly worse than the control technique, thus indicating its versatility for all the considered classifiers.

The selected base filter does not determine the good or bad behavior of the filtering process. For instance, while SW-IPF offers an outstanding performance, NN-IPF is not able to achieve the same behavior. When considering to EF as the base filter, NN-EF is not able to filter as accurately as SW-EF, indicating that the nearest neighbor technique is not well suited to this base filter or RNG, which also performs worse when used in the NN wrapper. Finally, we must mention that the combination of several filters is not a good choice either.

7. Instances removed per filter

While in the previous section we have focused on the performance impact of applying the proposed filters to eliminate negative instances within the bags, in this section we will study how these improvements relate with the amount of instances removed per each filter. Standard assumption in MIL hold that only a single positive instance is necessary to label the entire bag. Thus, removing other instances from the bag would help to obtain a more accurate description of the label itself, maybe rewarding aggressive filters.

Fig. 2 depicts the average amount of instances removed per each filter in each noise level. We observe that the amount of the average number of filtered instances vary greatly among the approaches. We may notice that the filtering strategy, SW or NN, has little impact on the amount of instances eliminated, but such a behavior is mainly dictated by the base filter used. For instance, RNG is an aggressive instance filter that removes a high number of instances. This behavior has been already observed in other related studies [29] and may cause the poor performance of NN-RNG: the over-filtering that RNG causes will remove positive instances from the bag that help to identify its label. Please note that SW-RNG is not as aggressive as NN-RNG, but still removes too many instances.

On the other hand, as we can observe in Fig. 2, IPF is the most conservative base filter. From the analysis in the previous sections, we also stated that SW-IPF is the best performing filter. By relating such good behavior with the amount of removed instances, SW-IPF cam be considered a balanced approach that removes a percentage of instances in the interval $[12, 22]$%. This amount is the approximate mid-point with respect to the percentages of MW and NN-IPF, being the most and least aggressive filters respectively. This successful balanced behavior provides two important hints when cleaning the intra-bag noise:

Table 5
Accuracy results for the different MIL classifiers after applying the noise filtering techniques. Best average result for each classifier and noise level is stressed in bold.

	SimpleMI	MITI	MILR	CitationKNN	MIBoost
0% noise					
NF	**81.3605**	77.4718	79.0508	**72.4096**	77.9908
SW-EF	79.2204	**78.9251**	79.3228	70.5946	78.9641
SW-IPF	80.5469	78.1800	78.7053	71.0911	**81.7616**
SW-RNG	78.4593	76.3976	77.5097	70.4366	78.7432
NN-EF	72.7608	73.6568	73.7986	65.2045	73.6199
NN-IPF	78.8171	77.1101	**79.3617**	71.8459	78.8491
NN-RNG	69.9011	71.9679	70.0943	60.2924	70.5762
MW	68.4639	71.4899	69.6220	60.3371	69.5003
5% noise					
NF	**80.3384**	76.6913	78.9554	71.5890	78.7612
SW-EF	77.6164	**77.4889**	78.5689	70.1127	79.0222
SW-IPF	80.1162	77.4428	78.1244	70.0040	**80.9387**
SW-RNG	78.2087	77.1990	77.7499	70.2344	78.7501
NN-EF	72.0620	72.4508	73.5859	64.9949	73.8404
NN-IPF	78.3623	76.0166	**79.1798**	**72.0762**	79.3059
NN-RNG	69.7148	71.5022	70.2754	60.0245	70.7421
MW	68.8596	70.4046	69.0959	60.1741	69.9353
10% noise					
NF	78.3608	74.9637	78.0317	69.7762	77.8180
SW-EF	76.0307	77.1759	76.1376	69.0320	78.7806
SW-IPF	**79.8879**	77.2747	**78.5287**	**70.4955**	**80.6538**
SW-RNG	78.2395	**77.8612**	77.4804	70.2023	77.9253
NN-EF	69.7500	70.8044	70.7473	63.5300	73.3847
NN-IPF	78.4198	75.5215	77.1174	70.1674	78.6762
NN-RNG	68.1038	68.9722	69.4192	59.8296	69.5985
MW	68.2312	68.8837	67.2414	60.3938	69.2206
15% noise					
NF	78.4201	72.3480	76.5043	69.1492	77.2428
SW-EF	74.6524	76.7261	76.6247	68.9719	76.5968
SW-IPF	**79.2590**	**77.1180**	**78.8736**	**70.5464**	**80.4784**
SW-RNG	77.8274	76.9029	77.2670	70.1241	77.9913
NN-EF	68.8008	70.9612	70.2904	63.3435	72.9865
NN-IPF	77.1075	75.2210	76.8067	69.8081	78.6054
NN-RNG	66.8640	69.1302	68.5310	60.0920	69.3150
MW	65.4844	68.2144	66.0521	60.6745	68.5245
20% noise					
NF	75.9204	69.1155	75.3476	67.9075	77.3866
SW-EF	72.5281	76.2914	74.6974	67.5939	77.3676
SW-IPF	**78.3870**	76.4976	**78.2482**	69.6947	**80.1285**
SW-RNG	78.2395	**77.8612**	77.4804	**70.2023**	77.9253
NN-EF	64.9703	69.1849	68.2730	60.5141	71.2693
NN-IPF	74.8545	72.6955	75.2751	68.2820	76.9014
NN-RNG	64.7350	67.8457	66.5601	58.6045	67.6538
MW	64.7910	67.3017	65.1009	59.0818	67.0583

- Even when the standard assumption holds, removing too many instances (possibly affecting positive instances as well) will hinder the classifier's performance. Keeping as many positive instances as possible is a feature that a good filter should exhibit.
- The average number of instances removed by good filters is more than the amount of artificially instances introduced in the bags. Since the application of the filters is beneficial, the bags should naturally contain a significant amount of negative examples that can be removed to improve the classification rate.

The reader may notice that these two objectives are contradictory, as we want to remove as many negative instances as we can before diminishing the relationship with the bag label given by the set of positive examples. Since the true distribution of positive instances is unknown, the more instances that are removed, the greater the probability of eliminating positive instances, thus the two points are the complete opposite.

From the results presented in this section, we may conclude that, among the algorithms investigated in this paper, SW-IPF is the best approach both in terms of performance and accurate negative instance removal.

Table 6
Friedman results for each classifier with Holm's post hoc test. **Control** denotes the best ranked algorithm, a '+' indicates a $p-value \leqslant 0.05$, a '*' denotes a $p-value \leqslant 0.1$ and a '=' is used to indicate a $p-value > 0.1$. Colors are adjusted to help identify the cases where the control algorithm is significantly better than the others.

Noise level	0%	5%	10%	15%	20%
MILR					
No filter	=	=	=	=	=
SW-EF	**Control**	**Control**	+	=	+
SW-IPF	=	=	**Control**	**Control**	**Control**
SW-RNG	+	+	+	=	=
NN-EF	+	+	+	+	+
NN-IPF	=	=	=	=	=
NN-RNG	+	+	+	+	+
MW	+	+	+	+	+
CitationKNN					
No filter	**Control**	=	=	=	=
SW-EF	=	=	=	=	+
SW-IPF	=	=	**Control**	**Control**	=
SW-RNG	=	=	=	=	**Control**
NN-EF	+	=	+	+	+
NN-IPF	=	**Control**	=	=	=
NN-RNG	+	+	+	+	+
MW	+	+	+	+	+
MIBoost					
No filter	+	+	=	+	=
SW-EF	+	+	+	+	=
SW-IPF	**Control**	**Control**	**Control**	**Control**	**Control**
SW-RNG	+	+	=	+	=
NN-EF	+	+	+	+	+
NN-IPF	+	+	=	+	+
NN-RNG	+	+	+	+	+
MW	+	+	+	+	+
MITI					
No filter	=	=	+	+	+
SW-EF	=	**Control**	**Control**	**Control**	=
SW-IPF	**Control**	=	=	=	=
SW-RNG	=	=	=	=	**Control**
NN-EF	=	=	+	*	+
NN-IPF	=	=	=	=	+
NN-RNG	=	=	+	=	+
MW	*	*	+	+	+
SimpleMI					
No filter	**Control**	=	=	=	*
SW-EF	=	*	+	*	+
SW-IPF	=	**Control**	**Control**	**Control**	**Control**
SW-RNG	+	+	+	=	=
NN-EF	+	+	+	+	+
NN-IPF	=	=	=	=	=
NN-RNG	+	+	+	+	+
MW	+	+	+	+	+

8. Concluding remarks

This paper tackles the problem of noise within the bags in MIC. While the standard assumption indicates that negative instances that do not contribute to the actual bag label may be present in the bags, analysis on whether the quantity or addition of such negative instances have an adverse impact on the knowledge extracted are still pending.

To alleviate the problem of having additional or transferred negative instances in the bags, we propose the application of noise filters to eliminate allegedly negative examples. In order to do so, two main approaches are carried out. On the one hand, we transform the MIL problem to a standard classification problem, where state-of-the-art techniques have demonstrated their good behavior. We then apply a base class label filter and finally rebuild the MIL problem without the noisy instances. On the other hand, we aim to exploit the similarity information between positive and negative bags. Thus, our goal is to help the base filters to identify negative examples by providing examples from the negative bags.

By studying the results in terms of accuracy for a varied set of representative MIL classifiers, we have observed that the base standard filter has strong influence on the performance results. We have also noted that converting the MIL problem to a standard problem to deal with the noisy examples is a valid strategy to improve the results when the negative instances

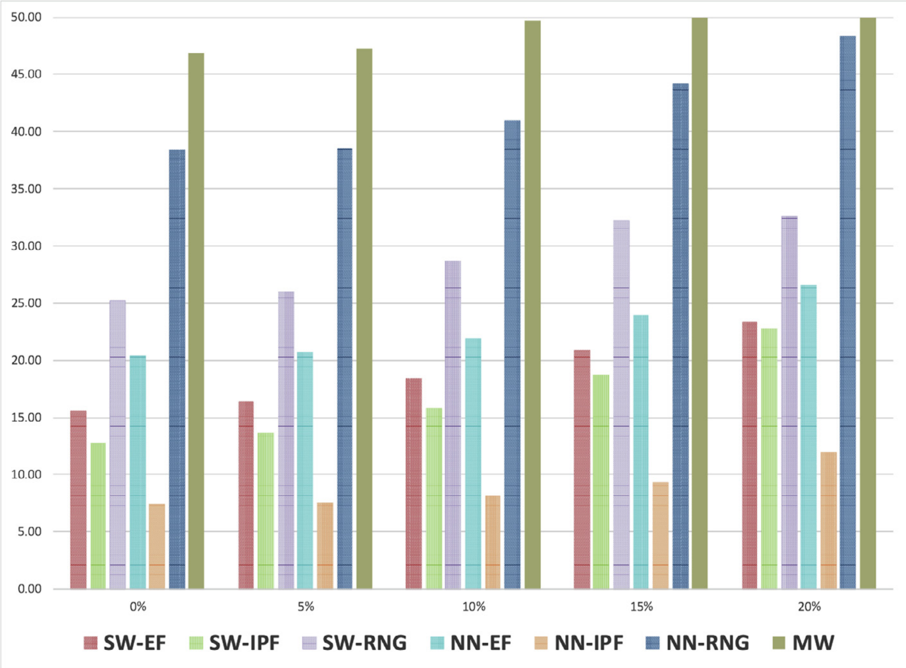

Fig. 2. Average amount of instances removed per filter in each noise level.

populate the bags. In particular, the combination of the Simple Wrapper strategy with IPF filter (SW-IPF) is the most successful technique, as it is able to safely work at 0% noise, and stands out from all the classifiers used to evaluate our proposal when noise is added. We have also shown that even at 0% induced noise, the removal of (negative) instances helps the classifiers to better identify the characteristics of positive bags.

This study opens the door to interesting questions and future efforts. Since noise filtering helps classifiers to improve the classification in MIL, it would be more efficient to create native MIL filters to deal with intra-bag noise, where the conversion to a standard classification problem is not needed. The exploitation of the information shared between positive bags also needs further analysis, as it could deliver better noise identification accuracy.

CRediT authorship contribution statement

Julián Luengo: Conceptualization, Software, Writing - original draft, Methodology, Writing - review & editing, Formal analysis, Validation. **Dánel Sánchez Tarragó:** Conceptualization, Software, Writing - original draft, Methodology. **Ronaldo C. Prati:** Formal analysis, Validation, Writing - review & editing, Supervision, Funding acquisition. **Francisco Herrera:** Validation, Writing - review & editing, Supervision, Project administration, Funding acquisition, Resources.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by project PID2020-119478GB-I00 granted by *Ministerio de Ciencia, Innovación y Univesidades*, project P18-FR-4961 by *Proyectos I+D+i Junta de Andalucia 2018* and the process nº 2015/20606-6, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

References

[1] E. Alpaydın, V. Cheplygina, M. Loog, D.M. Tax, Single-vs. multiple-instance classification, Pattern Recognition 48 (9) (2015) 2831–2838..
[2] J. Amores, Multiple instance classification: Review, taxonomy and comparative study, Artificial Intelligence 201 (2013) 81–105.
[3] H. Blockeel, D. Page, A. Srinivasan, Multi-instance tree learning, in: Proceedings of the 22nd International Conference on Machine Learning, ACM, 2005.
[4] C.E. Brodley, M.A. Friedl, Identifying mislabeled training data, Journal of Artificial Intelligence Research 11 (1999) 131–167.

[5] J. Chai, Z. Chen, H. Chen, X. Ding, Designing bag-level multiple-instance feature-weighting algorithms based on the large margin principle, Information Sciences 367 (2016) 783–808.
[6] Y. Chen, J. Bi, J.Z. Wang, Miles: Multiple-instance learning via embedded instance selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (12) (2006) 1931–1947.
[7] Y. Chevaleyre, J.-D. Zucker, Noise-tolerant rule induction from multi-instance data, in: ICML 2000, Workshop on Attribute-value and Relational Learning, 2000..
[8] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[9] T.G. Dietterich, R.H. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, Artificial Intelligence 89 (1–2) (1997) 31–71.
[10] F. Eibe, M. Hall, I. Witten, The Weka Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann..
[11] L. Fei, B. Zhang, C. Tian, S. Teng, J. Wen, Jointly learning multi-instance hand-based biometric descriptor, Information Sciences 562 (2021) 1–12.
[12] W. Feng, S. Boukir, Class noise removal and correction for image classification using ensemble margin, in: 2015 IEEE International Conference on Image Processing (ICIP), IEEE, 2015.
[13] B. Frénay, M. Verleysen, Classification in the presence of label noise: A survey, IEEE Transactions on Neural Networks and Learning Systems 25 (5) (2014) 845–869.
[14] C. Fu, R. Duan, E. Kayacan, Visual tracking with online structural similarity-based weighted multiple instance learning, Information Sciences 481 (2019) 292–310.
[15] D. Gamberger, R. Boskovic, N. Lavrac, C. Groselj, Experiments with noise filtering in a medical domain, in: Proceedings of the Sixteenth International Conference on Machine Learning, Morgan Kaufmann Publishers, 1999.
[16] D. Gamberger, N. Lavrac, S. Dzeroski, Noise Detection and Elimination in Data Preprocessing: experiments in medical domains, Applied Artificial Intelligence 14 (2000) 205–223.
[17] L.P. Garcia, J.A. Sáez, J. Luengo, A.C. Lorena, A.C. de Carvalho, F. Herrera, Using the one-vs-one decomposition to improve the performance of class noise filters via an aggregation strategy in multi-class classification problems, Knowledge-Based Systems 90 (C) (2015) 153–164.
[18] L.P.F. Garcia, A.C.P.L.F. de Carvalho, A.C. Lorena, Effect of label noise in the complexity of classification problems, Neurocomputing 160 (2015) 108–119.
[19] 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.
[20] K. He, W. Zhao, X. Xie, W. Ji, M. Liu, Z. Tang, Y. Shi, F. Shi, Y. Gao, J. Liu, et al, Synergistic learning of lung lobe segmentation and hierarchical multi-instance classification for automated severity assessment of covid-19 in ct images, Pattern Recognition 113 (2021) 107828.
[21] M.A. Hernández, S.J. Stolfo, Real-world data is dirty: data cleansing and the merge/purge problem, Data Mining and Knowledge Discovery 2 (1998) 9–37.
[22] J. Hernández-González, I. Inza, J.A. Lozano, Weak supervision and other non-standard classification problems: a taxonomy, Pattern Recognition Letters 69 (2016) 49–55.
[23] F. Herrera, S. Ventura, R. Bello, C. Cornelis, A. Zafra, D. Sánchez-Tarragó, S. Vluymans, Multiple instance learning, in: Multiple Instance Learning, Springer, 2016, pp. 17–33..
[24] T.M. Khoshgoftaar, P. Rebours, Improving software quality prediction by noise filtering techniques, Journal of Computer Science and Technology 22 (2007) 387–396.
[25] E.Ş. Küçükaşcı, M.G. Baydoğan, Bag encoding strategies in multiple instance learning problems, Information Sciences 467 (2018) 559–578.
[26] T. Leung, Y. Song, J. Zhang, Handling label noise in video classification via multiple instance learning, in: 2011 IEEE International Conference on Computer Vision (ICCV), IEEE, 2011.
[27] Y. Li, D.M.J. Tax, R.P.W. Duin, M. Loog, Multiple-instance learning as a classifier combining problem, Pattern Recognition 46 (3) (2013) 865–874, https://doi.org/10.1016/j.patcog.2012.08.018.
[28] J. Luengo, D. Sánchez-Tarragó, R.C. Prati, F. Herrera, A first study on the use of noise filtering to clean the bags in multi-instance classification, in: Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications, LOPAL '18, Association for Computing Machinery, New York, NY, USA, 2018..
[29] J. Luengo, S.-O. Shim, S. Alshomrani, A. Altalhi, F. Herrera, Cnc-nos: Class noise cleaning by ensemble filtering and noise scoring, Knowledge-Based Systems 140 (2018) 27–49.
[30] O. Maron, Learning from ambiguity, Ph.D. thesis, Massachusetts Institute of Technology, 1998..
[31] D.F. Nettleton, A. Orriols-Puig, A. Fornells, A study of the effect of different types of noise on the precision of supervised learning techniques, Artificial Intelligence Review 33 (4) (2010) 275–306.
[32] R.C. Prati, J. Luengo, F. Herrera, Emerging topics and challenges of learning from noisy data in nonstandard classification: a survey beyond binary class noise, Knowledge and Information Systems (2018) 1–35.
[33] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1993.
[34] Z. Ren, J. Han, N. Cummins, Q. Kong, M.D. Plumbley, B.W. Schuller, Multi-instance learning for bipolar disorder diagnosis using weakly labelled speech data, in: Proceedings of the 9th International Conference on Digital Public Health, 2019.
[35] J.A. Sáez, M. Galar, J. Luengo, F. Herrera, INFFC: An iterative class noise filter based on the fusion of classifiers with noise sensitivity control, Information Fusion 27 (2016) 19–32.
[36] J. Sánchez, F. Pla, F. Ferri, Prototype selection for the nearest neighbor rule through proximity graphs, Pattern Recognition Letters 18 (1997) 507–513.
[37] D.J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, CRC Press, 2003.
[38] B. Sluban, D. Gamberger, N. Lavrač, Ensemble-based noise detection: noise ranking and visual performance evaluation, Data Mining and Knowledge Discovery 28 (2) (2014) 265–303.
[39] P. Sudharshan, C. Petitjean, F. Spanhol, L.E. Oliveira, L. Heutte, P. Honeine, Multiple instance learning for histopathological breast cancer image classification, Expert Systems with Applications 117 (2019) 103–111.
[40] C.-M. Teng, Correcting Noisy Data, in: Proceedings of the Sixteenth International Conference on Machine Learning, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1999..
[41] F. Vázquez, J.S. Sánchez, F. Pla, A stochastic approach to wilson's editing algorithm, in: J.S. Marques, N. Pérez de la Blanca, P. Pina (Eds.), Pattern Recognition and Image Analysis: II Iberian Conference, IbPRIA 2005, Estoril, Portugal, June 7–9, 2005, Proceedings, Part II, 2005..
[42] S. Verbaeten, A.V. Assche, Ensemble methods for noise elimination in classification problems, in: Fourth International Workshop on Multiple Classifier Systems, Springer, 2003..
[43] J. Wang, J.-D. Zucker, Solving multiple-instance problem: A lazy learning approach..
[44] N. Weidmann, E. Frank, B. Pfahringer, A two-level learning method for generalized multi-instance problems, in: European Conference on Machine Learning, Springer, 2003..
[45] D. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Transactions on Systems and Man and Cybernetics 2 (3) (1972) 408–421.
[46] X. Zeng, T. Martinez, Using decision trees and soft labeling to filter mislabeled data, Journal of Intelligent Systems 17 (4) (2008) 331–354.
[47] S. Zhong, T.M. Khoshgoftaar, N. Seliya, Analyzing software measurement data with clustering techniques, IEEE Intelligent Systems 19 (2) (2004) 20–27.
[48] X. Zhu, X. Wu, Class noise vs. attribute noise: a quantitative study, Artificial Intelligence Review 22 (2004) 177–210.