Expert Systems with Applications 42 (2015) 5065–5075

Contents lists available at ScienceDirect

Expert Systems with Applications

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

# A biclustering approach for classification with mislabeled data

Fabrício O. de França [a], André L.V. Coelho [b,*]

[a] Center of Mathematics, Computing and Cognition (CMCC), Federal University of ABC (UFABC), Brazil
[b] Graduate Program in Applied Informatics, Center of Technological Sciences, University of Fortaleza (UNIFOR), Brazil

ARTICLE INFO

*Article history:*
Available online 6 March 2015

*Keywords:*
Classification
Noisy labels
Biclustering
Feature learning

ABSTRACT

Labeling samples on large data sets is a demanding task prone to different sources of errors. Those errors, denoted as noise, can significantly impact the performance of a classification algorithm due to overfitting of wrongly labeled data. So far, this problem has been treated by avoiding the overfitting and correcting mislabeled data through similarity analysis. The former approach can be affected by the curse of dimensionality and some mislabeled data will not be corrected. In this paper, we investigate the use of a biclustering approach to capture local models of coherence across subsets of instances and attributes. Those models are used to replace and augment the attributes of the original dataset. Through a systematic series of experiments, we have assessed the performance of the proposed approach, referred to as *BicNoise*, by considering different rates and types of label noise, and also different types of classifiers, binary datasets, and evaluation metrics. The good results achieved suggest that the transformed data can alleviate the dimensionality problem, reduce the redundancy of correlated features and improve the separability of the data, thus improving the classifier performance (most noticeably, in the highest noise settings).

## 1. Introduction

The investigation of the effects of noisy data on the performance of classification algorithms is a research line that has attracted a great deal of interest in machine learning (Atla, Tada, Sheng, & Singireddy, 2011; Nettleton, Orriols-Puig, & Fornells, 2010; Wu & Zhu, 2008; Zhu, Wu, Khoshgoftaar, & Shi, 2007). This is because datasets derived from real-world problems are usually plagued with several types of noise, bringing much uncertainty to the classifier induction process. The noise due to data mislabeling, in particular, which entails the modification (either random or not) of the observed labels assigned to the data instances (objects), can be potentially harmful and very difficult to cope with, since it can severely misconfigure the underlying relationships between the input (instance) and output (class) spaces (Frénay & Verleysen, 2014; Zhu & Wu, 2004).

A large body of work on the topic of supervised classification with label noise has emerged in the preceding years (Frénay & Verleysen, 2014). On one hand, there are approaches aiming at improving the quality of the noisy training data by modeling, detecting and then correcting, or simply removing, the affected cases. These methods are usually referred to as data cleansing methods (Brodley & Friedl, 1999; Guan, Yuan, Lee, & Lee, 2011; Zhu, Wu, & Chen, 2006). On the other hand, there are approaches, called noise-tolerant (or noise-robust), that can deal intrinsically with label noise while inducing the classifier models and, thus, do not depend on data preprocessing (Abellán & Moral, 2003; Abellán & Masegosa, 2012; Bootkrajang & Kabán, 2012, 2014). Finally, there are also some approaches, referred to here as hybrid ones, which combine features and properties of the abovementioned classes, e.g. by creating probabilistic models of label noise and then using this information to improve the noise-tolerance of the classifier during its training (Bouveyron & Girard, 2009; Rebbapragada & Brodley, 2007; Tabassian, Ghaderi, & Ebrahimpour, 2012a, Tabassian, Ghaderi, & Ebrahimpour, 2012b; Wang et al., 2012).

In this paper, we report on an empirical study investigating a novel approach for tackling the label noise problem. The approach, referred to as *BicNoise*, centers on the notion of biclusters (Cheng & Church, 2000; Madeira & Oliveira, 2004), i.e., submatrices of the training dataset showing high coherence of values across subsets of instances, attributes, and possibly class labels. By resorting to the local correlation models captured by the biclusters, we show that it is possible to elicit (learn) good discriminative features for improving the generalization performance of the induced classifiers. Different BicNoise variants are presented and assessed, which vary according to the way they modify the original dataset as well as to whether the label information of the training instances is used or not (supervised/unsupervised modes). Through a systematic series of experiments, we have assessed the performance of

* Corresponding author.
*E-mail addresses:* folivetti@ufabc.edu.br (F.O. de França), acoelho@unifor.br (A.L.V. Coelho).

http://dx.doi.org/10.1016/j.eswa.2015.02.045
0957-4174/© 2015 Elsevier Ltd. All rights reserved.

the BicNoise variants by considering different rates and types of label noise, and also different types of classifiers, binary datasets, and evaluation metrics.

The rest of the paper is structured as follows. In Section 2, we provide a brief survey on recent related work and a contrast on the label noise taxonomies independently conceived by Frénay and Verleysen (2014) and Rider, Johnson, Davis, Hoens, and Chawla (2013), which were both considered in the experiments reported in this work. Also in this section, we overview the main concepts associated with the biclustering task, giving special emphasis to the bicluster models and biclustering algorithm used in our experiments. In Section 3, we present in detail the BicNoise approach and its variants. Further, in Section 4, we outline the way the computational experiments were set up, and then present and discuss the several results achieved. Finally, Section 5 concludes the paper and provides remarks on future work.

## 2. Background

In the first subsection that follows, recent papers investigating the data mislabeling problem are overviewed. Then, two alternative classifications of the types of label noise are reviewed and contrasted. Finally, we focus specifically on the main concepts related to the biclustering task.

### 2.1. Related work

In a recent survey, Frénay and Verleysen (2014) have provided a comprehensive characterization of the problem of classification in the presence of label noise. Different definitions and sources of label noise were considered as well as several data cleansing, noise-tolerant and hybrid approaches were reviewed. Moreover, the authors have analyzed different statistical measures for validating the performance of algorithms within the label noise scenario.

Some noise-tolerant approaches for dealing with the data mislabeling problem are based on kernelized machines, such as the label-noise robust Kernel Logistic Regression classifier proposed by Bootkrajang and Kabán (2014). In this work, the authors have employed a multiple kernel learning setting, jointly with a Bayesian regularisation scheme, in order to determine the complexity parameters of the kernelized logistic regression models when no trusted validation set is available. Empirical results on 13 benchmark data sets and two real-world applications have demonstrated the success of the proposed approach.

Other approaches are based on the notion of classifier ensembles (Tabassian et al., 2012a; Guan, Yuan, Ma, & Lee, 2014). In this context, several classifiers or several variations of the same classifier are trained under the assumption that, for each mislabeled instance, only a minor set of the classifiers will learn their incorrect label through overfitting. Although interesting, one weakness of this approach has to do with the fact that whenever a mislabeled instance is located at the frontier of two or more classes, the majority of the classifiers will incorrectly learn a mistaken boundary for those classes.

Other approaches are variants of the Bagging and Boosting techniques (Abellán & Masegosa, 2012; Cantador & Dorronsoro, 2005; Cao, Kwong, & Wang, 2012). Here, the same classifier is trained by using different samples from the dataset, leading to different class boundaries that are combined afterwards. The main argument behind the Bagging schemes devised by Abellán and Masegosa (2012) in particular is that it is expected that each mislabeled instance will be part of just a few of the generated training sets, thus a majority of the boundaries will not be influenced by such instance. Even though the proposed schemes usually improve the performance, their success still strongly depends on which instances were mislabeled.

Finally, a more conceptually elaborated approach published in Expert Systems with Applications (Mantas & Abellán, 2014b; Mantas & Abellán, 2014a) makes use of the theory of Imprecise Probabilities, which deals with vague and conflicting information, for modeling the probability of each class. By using such theory, the authors have shown that the performance of the decision classifier C4.5 could be significantly improved when under the influence of noisy labels.

One should notice that the aforementioned approaches have two issues in common: (i) they do not entirely discard/transform the mislabeled data; and (ii) in most cases their performance is much dependent on the particular locations of the incorrect instances. These issues, however, do not appear in the approach investigated in the present paper, which aims at improving the separability of the classes by extracting novel features directly from the noisy data without resorting to any previous information (probabilistic or not) regarding the training set labels.

### 2.2. Statistical models of label noise

In most of the studies involving the classification of mislabeled instances, there is an implicit assumption that data are mislabeled completely at random. This assumption may be unrealistic in some real-world scenarios where multiple sources of systematic biases may happen during experimentation and data collection.

By mirroring the types of mechanisms usually considered in the missing value literature (Allison, 2002; Little & Rubin, 2002), Frénay and Verleysen (2014) and Rider et al. (2013) came up with two alternative taxonomies for modeling the different types of biases underlying the mislabeling process. According to the taxonomy of Frénay and Verleysen (2014), the three statistical models of label noise are defined as follows:

- *Noisy completely at random* (NCAR), whereby the mislabeling of an instance is viewed as a completely random process;
- *Noisy at random* (NAR), whereby the probability of mislabeling depends (solely) on the true class; and
- *Noisy not at random* (NNAR), whereby the mislabeling of an instance depends both on the true class and the particular values assumed by the instance attributes.

The authors emphasize that the first model can be considered as a special case of the second, while the third model is at the same time the more generic, complex, and realistic one.

On the other hand, the taxonomy adopted by Rider et al. (2013) was devised having in mind binary classification problems with a poorly defined negative class; that is, problems where it is known beforehand that only one of the classes can have its label flipped. The three types of biases accounting for the mislabeling of the positive class instances were defined as:

- *Biased completely at random* (BCAR), whereby the label change of the positive class occurs uniformly at random;
- *Biased at random* (BAR), whereby the mislabeling can be (completely) explained within the data (e.g. due to some property of the class and/or an explicit attribute of the dataset); and
- *Biased not at random* (BNAR), whereby the mislabeling happens as a consequence of some aspect not explicitly available in the dataset (e.g. due to some property of a latent attribute).

One should notice that this second taxonomy takes into account the possibility of having labeling errors due to external (latent) factors.

### 2.3. Biclustering

Biclustering, also known as co-clustering, is the process of finding subsets of rows and columns of a given data matrix with elements expressing high correlation (Cheng & Church, 2000). This data matrix may represent different kinds of numerical data in the form of objects (rows) and their attributes (columns).

Since its popularization, many researchers have applied this concept to handle problems that require the extraction of meaningful subgroups of the main dataset. The main motivation is usually to find data instances that are correlated under only a subset of the attributes, something that is above the capabilities of usual clustering methods. Some examples of biclustering applications include dimensionality reduction (Agrawal, Gehrke, Gunopulus, & Raghavan, 1998), information retrieval and text mining (de Castro, de França, Ferreira, & Von Zuben, 2007; Dhillon, 2001; de França, 2012), collaborative filtering (Symeonidis, Nanopoulos, Papadopoulos, & Manolopoulos, 2008), biological data analysis (Madeira & Oliveira, 2004), and missing data estimation (de França, Coelho, & Zuben, 2013).

As each application may have specific requirements, many different algorithms have been proposed to find biclusters for different purposes (Cheng & Church, 2000; Agrawal et al., 1998; de Castro et al., 2007; Coelho, de França, & Von Zuben, 2009; de França & Von Zuben, 2010). Since the term *biclustering* loosely refers to a group of data analysis approaches, and it is a ill-defined problem *per se*, those approaches differ on how they formulate the biclustering search problem and, thus, on how different types of biclusters are elicited.

One taxonomy of biclustering algorithms is related to how they are evaluated. In this context, an evaluation measure can be thought of as a local similarity measure between objects on specific regions of the data matrix. For instance, some algorithms search for constant values within the biclusters, while others try to find patterns that present some form of coherence (consistence) within the values of the elements (Madeira & Oliveira, 2004).

The coherence approach is particularly interesting because it can generalize several other types and represent a subset of data by means of as few as three variables. The additive coherence assumes that a subset of data presents a profile identical to the one exhibited by other rows or columns, except for a constant bias. In this case, each element $a_{i,j}$ of the submatrix can be calculated as (Cheng & Church, 2000):

$$a_{i,j} = a_{I,j} + a_{i,J} - a_{I,J}, \tag{1}$$

where $a_{I,j}$ is the mean value of the $j$th column of the bicluster, $a_{i,J}$ is the mean value of the $i$th row, and $a_{I,J}$ is the mean value of the bicluster.

Finding a coherent bicluster is the same as finding a bicluster that minimizes the error between the calculated and the real values of an element of the matrix. Then, the mean squared residue (MSR) $H(I,J)$ for the additive coherence becomes

$$H(I,J) = \frac{1}{|I||J|}\sum_{i,j \in I,J} r_{i,j}^2, \tag{2}$$

where $|I|$ is the total number of rows of the bicluster, $|J|$ is the total number of columns of the bicluster, and $r_{i,j}$ is given by

$$r_{i,j} = a_{i,j} - a_{I,j} - a_{i,J} + a_{I,J}. \tag{3}$$

Due to the presence of noise in most real-world data sets, the minimization of the MSR score should be constrained by a threshold $\delta$ (Cheng & Church, 2000). The choice of a proper value for $\delta$ may not be trivial as each dataset may present a different nature in terms of its attribute measurements. Another measure that can be used together with MSR is the size of the bicluster, often called as *volume*. Under this perspective, a good bicluster can be defined as the one that minimizes noise (MSR) while maximizing information (volume) (Coelho et al., 2009). So, an effective calibration rule for the value of $\delta$ is to find the minimum value which defines biclusters large enough to contain any information.

A recent swarm intelligence based approach was introduced by de França and Von Zuben (2010), named as SwarmBCluster, which could outperform other heuristic/exact biclustering algorithms on both MSR and volume criteria. Specifically, SwarmBCluster is a bioinspired metaheuristic based on an ant colony optimization (ACO) algorithm originally proposed by Dorigo, Bonabeau, and Theraulaz (2000). Among other functionalities, this algorithm is capable of uncovering several biclusters of varying volumes in a single run employing any type of coherence measure as fitness function, as empirically shown by de França and Von Zuben (2011). Moreover, as evidenced in the experiments with missing data conducted in de França et al. (2013), SwarmBCluster has shown high levels of robustness to noise while eliciting coherent biclusters. The reader is referred to the aforementioned papers for a full description of the main steps behind this algorithm.

In another recent work of our group (Prati & de França, 2013), SwarmBCluster was used in the context of multilabel classification. The general idea was that the set of discovered biclusters could yield a compact representation of the original dataset based on the local coherence they capture. The empirical results achieved on benchmark datasets showed a significant improvement of performance on most of the problems when compared to well-known multilabel approaches, leading to a conclusion that the extracted biclusters (viewed as novel attributes) hold significant information about the data that can be exploited to help the classification process. The BicNoise approach, introduced in this paper, is inspired by this strategy.

## 3. Dealing with mislabeled data with biclustering

In the last section, we mentioned that a bicluster can represent a local model for a subset of objects which express some form of correlation under a subset of attributes. These local models are generated in an unsupervised way and, thus, are unbiased regarding the object classes.

As such, each bicluster model can be seen as a new attribute that compacts the information captured by some of the original features, regarding specifically the selected objects. Notice that this strategy alone does not guarantee that two objects inside the same bicluster belong to the same class; in fact, there might exist *latent* classes grouping those objects together.

So, we can describe a given (training) dataset by the set of its local correlations captured by these models. Each elicited bicluster can thus give birth to a novel binary attribute vector, whose elements are active (i.e., value equal to ‘1’) only for those objects belonging to the bicluster.

The BicNoise approach proposed here makes direct use of this simple biclustering-induced quantization strategy in order to improve the classification task when the labels are not always correct due to the presence of noise. Given a dataset comprised of a labeled set of instances $T_r$, contaminated by noise on the labels $C$, and an unlabeled test set $T$, used solely to measure the final performance of the induced classifiers, the transformed training and test sets are generated following the steps itemized in Algorithm 1 operating on the biclusters found by the SwarmBCluster algorithm.

**Algorithm 1:** Main steps of BicNoise

```
Data: the training set $T_r$ used to extract the biclusters, the
    test set $T$, the set of class labels $C$, the MSR threshold $\delta$.
Result: the modified training $T'_r$ and test $T'$ datasets.
$B$ = SwarmBCluster($T_r, \delta$);
for each object $t \in T$ do
    for each bicluster $b \in B$ do
        Insert $t$ into $b$;
        if $MSR(b) > \delta$ then
            Remove $t$ from $b$;
$T'_r, T'$ = ModifyDataset ($T_r, T, C, B$);
```

More specifically, SwarmBCluster is performed on the training set in order to find a set of biclusters with MSR values less than $\delta$. After generating the biclusters set $B$, each instance from the test partition of the dataset is temporarily checked against each bicluster from $B$ in order to verify whether this object could be included into the bicluster without raising its MSR value above $\delta$. In other words, a test instance $i$ is said to be *compatible* with a bicluster $b$ if the insertion of $i$ into $b$ would not affect the levels of coherence already captured by $b$. It should be said that this "compatibility check" is made independently for each test instance and is used solely to transform the test set (it does not alter the bicluster models in any way). A bicluster augmented with compatible test instances will be referred to as an *augmented bicluster* in the sequel.

Finally, the augmented biclusters set is used to alter the original dataset, for both train and test partitions, through the function *ModifyDataset*(). This routine should use the information available at the biclusters set $B$ in order to create a more informative set of features that helps to reduce the negative effect of misclassified instances.

For the sake of exploring different ways of data manipulation, we have proposed three variants of *ModifyDataset*(), which rely on slightly different information:

- *BicFeat:* In this variant, the whole feature set is replaced by the binary features induced by the augmented biclusters set. Each instance $T_i$ will have $|B|$ features and the $j$th feature will indicate whether this instance belongs to the $j$th bicluster or not.
- *BicExt:* This variant will extend the original features set by appending the novel features inferred by the BicFeat approach.
- *BicClass:* With this variant the original features set is extended by appending only those features of BicFeat representing high purity biclusters.

The underlying hypothesis here is that the simple transformation of the original dataset through BicFeat leads to a simplified description of the data evidencing those attributes that better separate the classes. However, since this simplification may lead to information loss, the BicExt variant was devised to maintain the original set of features but reinforcing their discriminative information with the insertion of the novel binary features set. On the other hand, since there is no guarantee that a given bicluster will select a set of features covering unique classes, the augmented bicluster set is filtered by BicClass in order to retain only those biclusters that are homogeneous with respect to the classes.

It is worth emphasizing that none of these schemes violate the premise that classifiers should be induced without any reference to test data since only the labels of the training data are eventually used. The main steps of BicClass are summarized in Algorithm 2, whereas a more operational description of the three variants is provided next.

The first two versions of *ModifyDataset*() extract a new set of binary features whereby, for each object $i$, a vector $\mathbf{f}$ of size $|B|$ is created in such a way that the element $f_{i,b} = 1$ if the object $i$ belongs to the bicluster $b$. BicFeat simply replaces the original feature set with the bicluster induced one, whereas in BicExt the new feature set is simply aggregated. BicFeat can be viewed as a data transformation technique akin to sparse autoencoders (Coates, Ng, & Lee, 2011) and other deep learning techniques (Hinton, Osindero, & Teh, 2006), seeking for a different representation that captures local correlations. Thus, although the cardinality of $B$ can be very high (which depends on the number of biclusters elicited in the first step), since the new attributes tend to be more discriminative than the original ones their use should not incur curse of dimensionality problems.

Regarding the BicClass variant, this selects a subset of the augmented biclusters based on the purity of each bicluster with respect to the class labels of the training set. To measure the information of each bicluster, we have resorted to the notion of entropy (Cover & Thomas, 2006), with the probability of each class being estimated as its relative frequency within the bicluster's set of objects. How this process happens for the case with two classes is illustrated in Algorithm 2.

**Algorithm 2:** Supervised strategy of BicClass to select the most informative biclusters

```
Data: the training set $T_r$, the test set $T$, the set of class labels $C$,
    the bicluster set $B$, thresholds $thr$ and $thrH$.
Result: the modified training $T'_r$ and test $T'$ datasets.
$B' = \emptyset$;
for each bicluster $b \in B$
    $TrRate = \frac{|\{t \in b \cap T_r\}|}{|b|}$;
    if $TrRate > thr$ then
        $p^+ = |\{t \in b \wedge C[t] = 1\}|$;
        $p^- = |\{t \in b \wedge C[t] = 0\}|$;
        $H = -(p^+ \log(p^+) + p^- \log(p^-))$;
        if $H < thrH$ then
            $B' = B' \cup b$;
$T'_r, T' = BicExt(T_r, T, B')$;
```

For each augmented bicluster, the strategy adopted by BicClass first verifies whether the ratio of training examples within the bicluster (*TrRate*) is higher than a given threshold (*thr*), so that the bicluster contains enough supervised information. As the second step, it calculates the entropy based on the probability of each class appearing in the given bicluster[1]; if this value is less than another threshold (*thrH*), this bicluster is included in the final set of selected biclusters ($B$). The values of *thr* and *thrH* were empirically set as 0.7 and 0.15, respectively. Finally, the original feature set is extended, such as in BicExt. Notice that the calculation of the entropy returns a smaller value when the bicluster's purity is higher. Moreover, it is worth emphasizing that the training set labels, noisy or not, are used only to filter out the biclusters and not to induce them.

As a final notice, the adopted biclustering algorithm, SwarmBCluster, was chosen mainly because of its characteristics of finding biclusters with a larger number of elements, higher coverage of data and lower MSR, as reported in de França and Von Zuben (2010). But any other biclustering algorithm, whether coherence-based or not, could be in principle adopted for the first step of BicNoise, such as those investigated in Cheng and Church (2000), Dhillon (2001), Dhillon, Mallela, and Modha (2003), Prelic et al., 2006. However, the study of the applicability of such algorithms is out of the scope of this paper. Also, the reader is referred to de França and Von Zuben (2010) for more details on the SwarmBCluster algorithm and its pros and cons.

[1] Since the normalization factor is common for the probabilities of both classes, it was removed for the sake of conciseness.

## 4. Computational experiments

In order to assess whether the BicNoise approach can leverage the classification performance under the influence of noisy labels, an extensive series of experiments has been conducted. In the next subsections, we give details on the experimental setup and the calibration of important control parameters of SwarmBCluster, and then we present and discuss the main results achieved.

### 4.1. Experimental setup

For the purpose of validation, four binary benchmark datasets were chosen[2]:

- *Wisconsin Diagnostic Breast Cancer:* 569 instances, 357 begnign and 212 malign, with 30 image characteristics extracted from breast mass.
- *Parkinsons' Disease:* 195 instances of 23 voice measurements from 145 patients with Parkinsons' disease and 52 healthy subjects.
- *Sonar Signals:* 208 patterns of sonar signals bounced off 111 metal cylinders and 97 rocks with a total of 60 different measures each.
- *Singh Prostate Cancer:* 339 measurements selected from a pool of 2153 gene expressions of 52 prostate tumors and 50 normal adjacent prostate tissue samples.

The choice of these datasets was motivated mostly by their attribute characteristics: all with real-valued features and having more than 20 attributes. The reasons are twofold: (i) SwarmBCluster works only with real valued data; and (ii) it would be meaningless to extract biclusters from a low dimensional dataset. The latter dataset, in particular, is very high dimensional in nature, and with a relatively very small set of samples. Moreover, in a recent systematic empirical study, Duch, Jankowski, and Maszczyk (2012) have shown that the second and third datasets of the previous list should be regarded as 'non-trivial' for simple, low-complexity classifiers.

For each of these datasets, the training process was performed on:

- the original data with no label noise;
- the data with noisy labels injected;
- the predicted training labels generated after training (in a first round) the classifier with the noisy labels;
- the novel dataset produced by BicFeat;
- the extended dataset produced by BicExt; and
- the extended dataset produced by BicClass.

Six well-known classification algorithms[3] were selected in order to assess the potentials of the different BicNoise variants (Witten & Frank, 2005): Multinomial Naïve Bayes (NB), CART (Tree), $k$-Nearest Neighbors (kNN), with $k = 1$, Support Vector Machines (SVM), Random Forest (RF) configured with 10 trees, and AdaBoost (Ada) configured with up to 100 trees. These classifiers represent different classes of algorithms, most of which have also been considered in other studies investigating empirically the effects of noise in classification (Guan et al., 2011; Nettleton et al., 2010; Rider et al., 2013). On what concerns the number of estimators of the ensemble models, this was set in preliminary experiments in a manner as to achieve the best average results on each dataset.

**Table 1**
Contrast between the label noise taxonomies considered in this work.

| Rider et al. (2013) | Frénay and Verleysen (2014) | Acronym used here |
|---|---|---|
| – | NCAR | NCAR |
| BCAR | NAR | NAR(a) |
| BAR | NNAR | NAR(b) |
| BNAR | – | NNAR |

Different types of label noise have been considered in this study, having in mind the two taxonomies discussed in Section 2.2. Table 1 provides a contrast between the label noise classes of these two taxonomies, indicating which classes can be regarded as similar in nature (at least partially) and how they will be referred to hereafter in this paper (third column).

Each type of noise was injected into the datasets via simple procedures as follows (Rider et al., 2013):

- *NCAR:* $X\%$ of both positive and negative labels were randomly chosen and flipped.
- *NAR(a):* $X\%$ of the positive labels were randomly chosen to become negative labels.
- *NAR(b):* the correlation between each pair of features is calculated, then the objects are sorted by the most uncorrelated feature and the first $X\%$ of the positive labels become negative.
- *NNAR:* the same process as in NAR(b) but with the uncorrelated feature removed from the dataset.

In order to verify the effects of degradation in classification performance due to the mislabeling process, we tested five different noise rates, namely $10\%, 20\%, 30\%, 40\%,$ and $50\%$. These noise rates were applied for each of the noise classes described above. Moreover, since the Parkinsons' Disease dataset is highly unbalanced, we also run experiments exchanging the positive with the negative class, denoting this version of the problem as *Park. (2)*.

All datasets were normalized via standardization before the application of the Bicnoise variants. This step is important to ease the calibration of the SwarmBCluster parameters. On the other hand, the performance results were produced via 10-fold stratified cross-validation (the same folds were used for all denoising methods) and assessed via well-known measures, namely Accuracy, Precision, Recall, F-measure, and AUC – the reader is referred to Witten and Frank (2005) for a formal definition of these measures.

### 4.2. Parameter setting of the biclustering algorithm

The values of the control parameters of SwarmBCluster (de França & Von Zuben, 2010) were empirically optimized in such a way that the biclusters unveiled were neither too small in volume nor with a high MSR value. The methodology adopted was to optimize each control parameter (e.g., $\delta$) separately, keeping the others (e.g., the bicluster model) all fixed. As a result, for all datasets but WDBC, the coherence metric employed as optimizing criterion was the additive coherence (Eq. (1)). For WDBC, better results were obtained by seeking out biclusters with approximately constant values along their columns (i.e., constant-column coherence (Madeira & Oliveira, 2004)). After proper calibration, the value of $\delta$ adopted in the experiments was 0.001 or 0.01, which depends on the dataset (refer to Fig. 1) but not to the type of label noise. Besides, five was the minimum number of objects allowed for a bicluster, whereas the minimum number of features pertaining to each bicluster varied for each dataset as follows: 10 for WDBC, 5 for Parkinsons, 35 for Sonar, and 20 for Singh. Finally, the total

[2] The first three datasets are available on the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml) while the latter can be downloaded from the supplementary material repository of reference (de Souto, Costa, de Araujo, Ludermir, & Schliep, 2008) (http://bioinformatics.rutgers.edu/Supplements/CompCancer/).

[3] These algorithms are available in the Scikit-Learn toolkit (http://scikit-learn.org/), which has been adopted as testbed for implementing and empirically assessing the BicNoise variants.

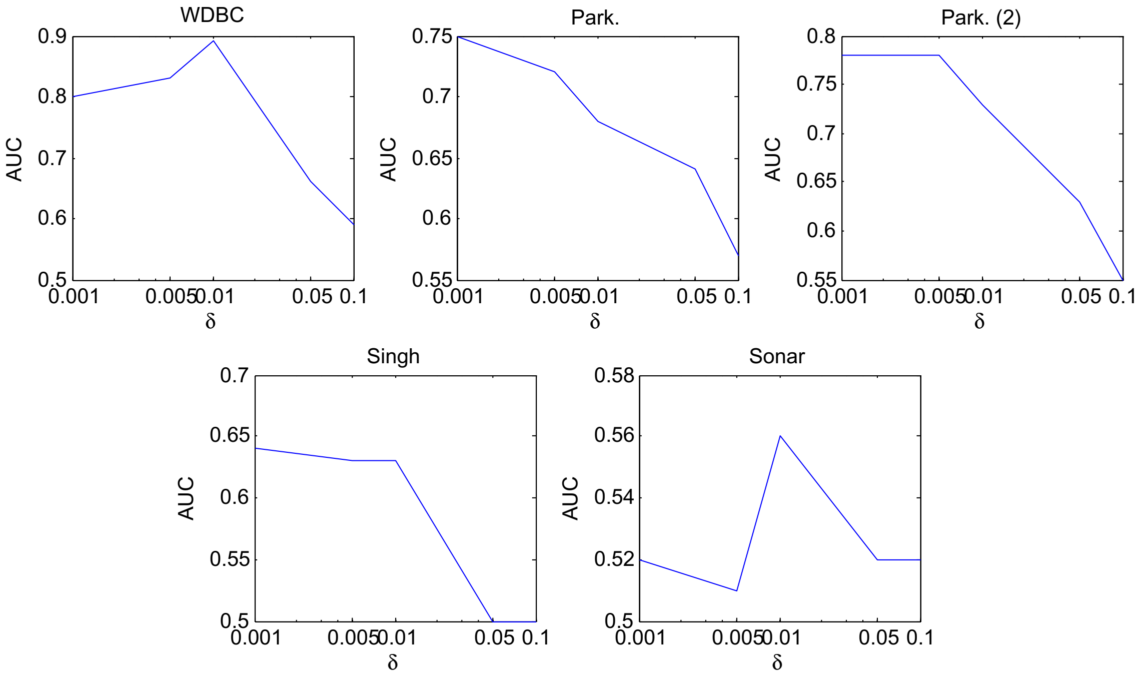

**Fig. 1.** Calibration of $\delta$ (in log-scale) for different datasets.

**Table 2**
Average number of biclusters generated by SwarmBCluster and average number of features for each BicNoise variant.

| Dataset | SwarmBCluster | BicFeat | BicExt | BicClass |
|---|---|---|---|---|
| WDBC | 400 | 400 | 430 | 230 |
| Park. | 383 | 383 | 405 | 213.5 |
| Park. (2) | 383 | 383 | 405 | 213.5 |
| Sonar | 532.5 | 532.5 | 592.5 | 326.25 |
| Singh | 1600 | 1600 | 1939 | 1139 |

amount of biclusters elicited by SwarmBCluster has varied according to each dataset – refer to Table 2, which also shows the average number of features generated for each BicNoise variant.

### 4.3. Results and discussion

Due to the extensive nature of the experiments, the main results achieved are summarized in the sequel in terms of AUC only. The rationale behind this choice has to do with the fact that a strong correlation was in general perceived between this measure and the others across all experiments – refer to Appendix A.

Tables 3–7 present the results delivered by the BicNoise variants for each of the noise rates separately. The second column of each table shows the denoising method. Besides the BicNoise variants, we also report results for three baseline methods, namely: (i) 'None', whereby the classifier is induced on training data free of any label noise; (ii) 'Noise', whereby the classifier is induced on training data polluted with label noise; and (iii) 'Class.', whereby the classifier is firstly trained on noisy data, then predicts the labels of these data, and finally is trained again on the relabeled data. In the third column, the classifier algorithm that worked best for the given denoiser is indicated. Finally, the subsequent columns show the average and standard deviation values of AUC achieved for each type of noise, followed by the rank of the denoiser in the contest within parentheses.

Additionally, we also recorded the AUC values of the induced classifiers on test data polluted with label noise. This helps to better evaluate the levels of generalization behind the induced hypotheses. Consider for instance the existence of two similar instances $i_1$ and $i_2$, the first belonging to the training partition and the second to the test partition. Without considering the possibility of label noise in test data, we could assess only the cases when both instances were noise-free or the first was mislabeled while the second was not. Conversely, by injecting noise in test data, it is also possible to assess the cases when both instances became corrupted or the second was mislabeled while the first was not.

For the sake of conciseness, we only report in the tables the sign of the difference in AUC between the non-noisy and noisy test cases within square brackets. A plus (negative) sign means that the classifier obtained better (worse) predictions for real labels than for the noisy labels in the test. This information is enough to indicate which denoising methods are more robust to label noise in unseen data.

On the other hand, in order to verify the statistical significance of the results, a paired Wilcoxon rank test (Demšar, 2006) with 0.05 confidence level was performed for each experimental setting (i.e., configuration of dataset, noise type and rate). Those experiments in which the best of the BicNoise variants have significantly prevailed or been outperformed by the best among 'Noise' or Class.' methods are highlighted in bold in the tables.

As expected, 'None' has systematically achieved the best results across all configuration settings, serving as a yardstick to measure the relative performance achieved by the denoising methods. Only with the low/moderate noise rates is that the denoisers could produce results somewhat equiparable to those produced by 'None', even though this observation does not hold equally for all datasets (contrast, for instance, WDBC and Park.). Once the noise rates are raised up to 50%, the differences in accuracy with respect to 'None' become more noticeable, which indicates that there is still much room for improvement in the design of denoising methods.

Contrasting the performance of BicNoise and non-BicNoise methods, in only 17 of the experimental settings the Wilcoxon test signaled a non-equivalence in performance between the best of the two classes, and this happened almost exclusively for the WDBC dataset (for the others, the methods have produced invariably large standard deviation values in accuracy, showing great overlap in the AUC values). In 65% of these cases, the best BicNoise variant prevailed, showing better performance when the noise rates are considerably high ($\geqslant 30\%$).

By considering the average AUC results alone, the contrast between the BicNoise and non-BicNoise methods also evidences that the latter usually outperform the former when low/moderate noise rates occur, while the converse is true for high noise rates. This can be certified by looking at the ranks of these methods across the tables and the aggregated ones summarized in Table 8. On the other hand, when considering the possibility of having label noise in test, BicFeat was the only approach that systematically obtained a positive difference between the real and noisy AUC values in two of the datasets considered, namely, WDBC and Singh. (In Sonar, only for 20% and 30% as noise rates, BicFeat did not achieve a positive difference.) Even though the average accuracy of BicFeat was usually not superior to the accuracy of the other BicNoise variants (see below), it was the single one capable of effectively reducing the bias imposed by the noisy labels in the training data and generalizing well to the test data.

These results conform with those obtained in de França et al. (2013) in which the biclustering approach was showed to be more robust against the noise artificially inserted into the dataset (but not on the class labels). This is due to the fact that biclustering discards the features that introduce noise under a subset of the objects in a way as to find subsets with high correlations. In the noisy label setting, in particular, the selective procedure of the biclustering algorithm helps to evidence that some mislabeled data are indeed more correlated to the opposite class than with the one that is currently being labeled.

Among the BicNoise variants, BicExt yielded the best average results in 50% of the configuration settings, BicClass came second with 37.7%, and BicFeat accounted for 12.3%. These results may indicate that the set of new features (elicited biclusters) show a complementary role to the set of original features. Although BicFeat performed worse than the other variants, this happens mostly for the Sonar and Singh datasets. However, once the label noise rate becomes more pronounced, the good performance of BicNoise in general also becomes more noticeable.

The reason for these latter results can be explained by the modus operandis of each BicNoise variant. Although unbiased to noise, the dataset generated by BicFeat, for instance, is less informative, since it discards the original attribute information. This is evidenced by noticing that its classification results are barely changed with the increase of noise. BicClass, on the other hand, yielded better results than BicFeat since it kept every information from the original data while complementing with only those biclusters considered more informative (homogeneous).

Considering the influence of the datasets, one can notice that all denoisers exhibit a higher degradation in performance for the latter two datasets, albeit for Singh the lag in performance with respect to 'None' becomes more salient. This may have to do with the "high dimensionality/small sample set" property of the Singh dataset. On the other hand, if one considers the effect of the noise type on the denoiser's performance, it seems that this factor depends on the dataset and noise rate considered. For instance, whereas for Sonar the accuracy of the methods was usually comparatively lower when coping with NAR(b) and NNAR for all noise rates, for Park., the effect of NAR(a) and NCAR was more remarkable for higher noise rates.

Finally, it is interesting to observe that for low/moderate levels of noise (10% to 30%) the best overall choice of the classifiers varied mostly among SVM, KNN, and the ensemble-based algorithms. Nevertheless, for the highest noise rate (50%), NB started to appear more frequently as the best model inducer. This result complies with what is generally reported in the literature (Nettleton et al., 2010): while NB has as strong points against noise the assumption

**Table 3**
Results with noise rate of 10%. Cases in which the statistical test indicated that the best BicNoise variant has significantly prevailed or been outperformed by the best non-BicNoise method are highlighted in bold.

| Dataset | | Alg. | NCAR | NAR(a) | NAR(b) | NNAR |
|---|---|---|---|---|---|---|
| WDBC | None | SVM | 0.98 0.06 (1) [+] | 0.98 0.06 (1) [+] | 0.98 0.06 (1) [+] | 0.98 0.06 (1) [+] |
| | Noise | SVM | **0.97 0.05** (**2**) [−] | 0.96 0.06 (2) [−] | 0.93 0.05 (3) [−] | 0.93 0.05 (3) [−] |
| | Class. | SVM | **0.98 0.08** (**1**) [−] | 0.96 0.06 (2) [−] | 0.94 0.06 (2) [−] | 0.94 0.06 (2) [−] |
| | BicClass | RF | **0.93 0.05** (**3**) [−] | 0.94 0.04 (3) [−] | 0.93 0.04 (3) [−] | 0.93 0.04 (3) [−] |
| | BicExt | RF | **0.93 0.05** (**3**) [−] | 0.94 0.04 (3) [−] | 0.92 0.04 (4) [−] | 0.93 0.04 (3) [−] |
| | BicFeat | NB | **0.87 0.06** (**4**) [+] | 0.89 0.06 (4) [+] | 0.88 0.06 (5) [+] | 0.87 0.05 (4) [+] |
| Park. | None | kNN | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] |
| | Noise | kNN | 0.80 0.22 (2) [+] | 0.85 0.20 (2) [+] | 0.87 0.20 (1) [+] | 0.87 0.19 (1) [+] |
| | Class. | kNN | 0.80 0.21 (2) [+] | 0.85 0.18 (2) [+] | 0.87 0.19 (1) [+] | 0.87 0.19 (1) [+] |
| | BicClass | Ada | 0.75 0.22 (4) [−] | 0.80 0.20 (4) [−] | 0.82 0.21 (2) [−] | 0.77 0.20 (3) [−] |
| | BicExt | Ada | 0.77 0.22 (3) [−] | 0.80 0.20 (4) [−] | 0.81 0.21 (3) [−] | 0.78 0.21 (2) [−] |
| | BicFeat | NB | 0.77 0.20 (3) [−] | 0.81 0.19 (3) [−] | 0.80 0.20 (4) [−] | 0.76 0.19 (4) [−] |
| Park. (2) | None | kNN | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] |
| | Noise | kNN | 0.86 0.20 (2) [−] | 0.83 0.21 (3) [−] | 0.82 0.22 (2) [−] | 0.82 0.22 (4) [−] |
| | Class. | kNN | 0.86 0.22 (2) [−] | 0.83 0.19 (3) [−] | 0.82 0.21 (2) [−] | 0.82 0.21 (4) [−] |
| | BicClass | SVM | 0.77 0.19 (4) [−] | 0.82 0.21 (4) [−] | 0.81 0.17 (3) [−] | 0.83 0.17 (3) [−] |
| | BicExt | SVM | 0.77 0.20 (4) [−] | 0.85 0.21 (2) [−] | 0.81 0.18 (3) [−] | 0.84 0.19 (2) [−] |
| | BicFeat | SVM | 0.78 0.19 (3) [+] | 0.82 0.15 (4) [+] | 0.79 0.19 (4) [+] | 0.81 0.18 (5) [+] |
| Sonar | None | Ada | 0.76 0.18 (1) [+] | 0.76 0.18 (1) [+] | 0.76 0.18 (1) [+] | 0.76 0.18 (1) [+] |
| | Noise | Ada | 0.74 0.18 (2) [−] | 0.76 0.16 (1) [−] | 0.68 0.17 (3) [−] | 0.69 0.17 (2) [−] |
| | Class. | Ada | 0.74 0.17 (2) [−] | 0.76 0.16 (1) [−] | 0.68 0.17 (3) [−] | 0.68 0.19 (3) [−] |
| | BicClass | Ada | 0.73 0.16 (3) [−] | 0.75 0.19 (2) [−] | 0.69 0.19 (2) [−] | 0.67 0.19 (4) [−] |
| | BicExt | Ada | 0.73 0.15 (3) [−] | 0.75 0.19 (2) [−] | 0.69 0.18 (2) [−] | 0.67 0.19 (4) [−] |
| | BicFeat | NB | 0.61 0.17 (4) [+] | 0.60 0.15 (3) [+] | 0.61 0.16 (4) [+] | 0.61 0.17 (5) [+] |
| Singh | None | SVM | 0.92 0.15 (1) [+] | 0.92 0.14 (1) [+] | 0.92 0.14 (1) [+] | 0.92 0.14 (1) [+] |
| | Noise | SVM | 0.92 0.14 (1) [+] | 0.90 0.14 (3) [+] | 0.90 0.14 (3) [+] | 0.90 0.15 (2) [+] |
| | Class. | SVM | 0.92 0.15 (1) [+] | 0.91 0.16 (2) [+] | 0.90 0.11 (3) [+] | 0.90 0.13 (2) [+] |
| | BicClass | SVM | 0.90 0.12 (2) [+] | 0.87 0.18 (4) [+] | 0.91 0.14 (2) [+] | 0.88 0.18 (3) [+] |
| | BicExt | SVM | 0.92 0.12 (1) [+] | 0.87 0.17 (4) [+] | 0.91 0.16 (2) [+] | 0.90 0.16 (2) [+] |
| | BicFeat | NB | 0.57 0.18 (3) [+] | 0.57 0.15 (5) [+] | 0.57 0.18 (4) [+] | 0.59 0.23 (4) [+] |

**Table 4**
Results with noise rate of 20%. Cases in which the statistical test indicated that the best BicNoise variant has significantly prevailed or been outperformed by the best non-BicNoise method are highlighted in bold.

| Dataset | | Alg. | NCAR | NAR(a) | NAR(b) | NNAR |
|---|---|---|---|---|---|---|
| WDBC | None | SVM | 0.98 0.06 (1) [+] | 0.98 0.06 (1) [+] | 0.98 0.06 (1) [+] | 0.98 0.06 (1) [+] |
| | Noise | SVM | **0.96 0.05** (**3**) [−] | **0.96 0.04** (**3**) [−] | **0.87 0.04** (**4**) [−] | 0.87 0.04 (4) [−] |
| | Class. | SVM | **0.97 0.06** (**2**) [−] | **0.97 0.05** (**2**) [−] | **0.88 0.05** (**3**) [−] | 0.87 0.05 (4) [−] |
| | BicClass | RF | **0.93 0.06** (**4**) [−] | **0.93 0.06** (**4**) [−] | **0.88 0.05** (**3**) [−] | 0.89 0.05 (2) [−] |
| | BicExt | RF | **0.91 0.07** (**5**) [−] | **0.91 0.05** (**5**) [−] | **0.89 0.05** (**2**) [−] | 0.89 0.04 (2) [−] |
| | BicFeat | NB | **0.90 0.06** (**6**) [+] | **0.87 0.07** (**6**) [+] | **0.88 0.06** (**3**) [+] | 0.88 0.05 (3) [+] |
| Park. | None | kNN | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.22 (1) [+] | 0.87 0.21 (1) [+] |
| | Noise | kNN | 0.79 0.19 (3) [−] | 0.75 0.17 (3) [−] | 0.82 0.20 (2) [−] | 0.82 0.22 (2) [−] |
| | Class. | kNN | 0.79 0.17 (3) [−] | 0.75 0.16 (3) [−] | 0.82 0.21 (2) [−] | 0.82 0.21 (2) [−] |
| | BicClass | Ada | 0.81 0.19 (2) [−] | 0.71 0.18 (5) [−] | 0.77 0.22 (4) [−] | 0.79 0.23 (3) [−] |
| | BicExt | NB | 0.81 0.18 (2) [−] | 0.78 0.15 (2) [−] | 0.79 0.21 (3) [−] | 0.79 0.24 (3) [−] |
| | BicFeat | NB | 0.78 0.14 (4) [−] | 0.74 0.16 (4) [−] | 0.77 0.20 (4) [−] | 0.78 0.21 (4) [−] |
| Park. (2) | None | kNN | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] |
| | Noise | kNN | 0.85 0.20 (2) [−] | 0.74 0.24 (3) [−] | 0.79 0.22 (4) [−] | 0.79 0.22 (4) [−] |
| | Class. | kNN | 0.85 0.21 (2) [−] | 0.74 0.23 (3) [−] | 0.79 0.23 (4) [−] | 0.79 0.23 (4) [−] |
| | BicClass | NB | 0.81 0.18 (3) [−] | 0.76 0.21 (2) [−] | 0.81 0.21 (3) [−] | 0.81 0.19 (2) [−] |
| | BicExt | NB | 0.79 0.17 (4) [−] | 0.74 0.22 (3) [−] | 0.82 0.21 (2) [−] | 0.77 0.21 (5) [−] |
| | BicFeat | NB | 0.77 0.17 (5) [+] | 0.74 0.21 (3) [+] | 0.81 0.17 (3) [+] | 0.80 0.18 (3) [+] |
| Sonar | None | Ada | 0.76 0.18 (1) [+] | 0.76 0.18 (1) [+] | 0.76 0.18 (1) [+] | 0.76 0.18 (1) [+] |
| | Noise | SVM | 0.70 0.13 (2) [−] | 0.66 0.15 (4) [−] | 0.66 0.19 (4) [−] | 0.63 0.19 (4) [−] |
| | Class. | RF | 0.68 0.14 (3) [−] | 0.67 0.14 (3) [−] | 0.67 0.21 (3) [−] | 0.63 0.21 (4) [−] |
| | BicClass | Ada | 0.68 0.18 (3) [−] | 0.70 0.17 (2) [−] | 0.70 0.19 (2) [−] | 0.67 0.17 (2) [−] |
| | BicExt | Ada | 0.68 0.16 (3) [−] | 0.70 0.14 (2) [−] | 0.62 0.18 (5) [−] | 0.66 0.17 (3) [−] |
| | BicFeat | NB | 0.58 0.14 (4) [−] | 0.57 0.14 (5) [−] | 0.60 0.15 (6) [−] | 0.60 0.15 (5) [−] |
| Singh | None | SVM | 0.92 0.14 (1) [+] | 0.92 0.15 (1) [+] | 0.92 0.14 (1) [+] | 0.92 0.14 (1) [+] |
| | Noise | SVM | 0.87 0.18 (3) [+] | 0.86 0.14 (3) [+] | 0.88 0.14 (2) [+] | 0.88 0.14 (2) [+] |
| | Class. | SVM | 0.89 0.16 (2) [+] | 0.86 0.14 (3) [+] | 0.88 0.14 (2) [+] | 0.88 0.15 (2) [+] |
| | BicClass | SVM | 0.80 0.16 (5) [+] | 0.90 0.14 (2) [+] | 0.87 0.15 (3) [+] | 0.88 0.18 (2) [+] |
| | BicExt | SVM | 0.82 0.15 (4) [+] | 0.90 0.12 (2) [+] | 0.87 0.15 (3) [+] | 0.88 0.16 (2) [+] |
| | BicFeat | Tree | 0.55 0.16 (6) [+] | 0.58 0.16 (4) [+] | 0.65 0.16 (4) [+] | 0.58 0.16 (3) [+] |

**Table 5**
Results with noise rate of 30%. Cases in which the statistical test indicated that the best BicNoise variant has significantly prevailed or been outperformed by the best non-BicNoise method are highlighted in bold.

| Dataset | | Alg. | NCAR | NAR(a) | NAR(b) | NNAR |
|---|---|---|---|---|---|---|
| WDBC | None | SVM | 0.98 0.06 (1) [+] | 0.98 0.06 (1) [+] | 0.98 0.06 (1) [+] | 0.98 0.06 (1) [+] |
| | Noise | SVM | **0.97 0.05** (**2**) [−] | **0.95 0.06** (**2**) [−] | **0.83 0.06** (**4**) [−] | **0.83 0.06** (**3**) [−] |
| | Class. | SVM | **0.97 0.06** (**2**) [−] | **0.93 0.06** (**3**) [−] | **0.83 0.06** (**4**) [−] | **0.83 0.06** (**3**) [−] |
| | BicClass | RF | **0.90 0.05** (**4**) [−] | **0.88 0.06** (**4**) [−] | **0.87 0.06** (**2**) [−] | **0.88 0.06** (**2**) [−] |
| | BicExt | RF | **0.91 0.06** (**3**) [−] | **0.88 0.06** (**4**) [−] | **0.87 0.07** (**2**) [−] | **0.88 0.06** (**2**) [−] |
| | BicFeat | NB | **0.89 0.06** (**5**) [+] | **0.88 0.06** (**4**) [+] | **0.86 0.06** (**3**) [+] | **0.88 0.07** (**2**) [+] |
| Park. | None | kNN | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] |
| | Noise | kNN | 0.77 0.18 (2) [−] | 0.76 0.19 (2) [−] | 0.80 0.20 (2) [−] | 0.80 0.20 (2) [−] |
| | Class. | kNN | 0.77 0.20 (2) [−] | 0.76 0.19 (2) [−] | 0.80 0.22 (2) [−] | 0.80 0.21 (2) [−] |
| | BicClass | SVM | 0.75 0.19 (4) [−] | 0.71 0.18 (4) [−] | 0.76 0.19 (4) [−] | 0.76 0.21 (3) [−] |
| | BicExt | SVM | 0.75 0.22 (4) [−] | 0.72 0.20 (3) [−] | 0.76 0.19 (4) [−] | 0.76 0.21 (3) [−] |
| | BicFeat | NB | 0.76 0.20 (3) [−] | 0.71 0.19 (4) [−] | 0.77 0.20 (3) [−] | 0.74 0.19 (4) [−] |
| Park. (2) | None | kNN | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] |
| | Noise | kNN | 0.79 0.21 (2) [−] | 0.76 0.22 (5) [−] | 0.75 0.21 (4) [−] | 0.75 0.26 (4) [−] |
| | Class. | kNN | 0.79 0.20 (2) [−] | 0.76 0.22 (5) [−] | 0.75 0.21 (4) [−] | 0.75 0.21 (4) [−] |
| | BicClass | NB | 0.77 0.20 (3) [−] | 0.80 0.20 (2) [−] | 0.76 0.22 (3) [−] | 0.77 0.22 (2) [−] |
| | BicExt | NB | 0.77 0.20 (3) [−] | 0.79 0.22 (3) [−] | 0.78 0.20 (2) [−] | 0.76 0.22 (3) [−] |
| | BicFeat | NB | 0.75 0.16 (4) [−] | 0.77 0.19 (4) [−] | 0.75 0.18 (4) [−] | 0.74 0.17 (5) [−] |
| Sonar | None | Ada | 0.76 0.18 (1) [+] | 0.76 0.18 (1) [+] | 0.76 0.18 (1) [+] | 0.76 0.18 (1) [+] |
| | Noise | SVM | 0.69 0.16 (3) [−] | 0.69 0.15 (2) [−] | 0.59 0.18 (3) [−] | 0.60 0.18 (2) [−] |
| | Class. | RF | 0.69 0.17 (3) [−] | 0.69 0.15 (2) [−] | 0.58 0.18 (4) [−] | 0.59 0.18 (3) [−] |
| | BicClass | Ada | 0.68 0.16 (4) [−] | 0.67 0.16 (3) [−] | 0.61 0.19 (2) [−] | 0.60 0.16 (2) [−] |
| | BicExt | Ada | 0.73 0.16 (2) [−] | 0.67 0.15 (3) [−] | 0.59 0.19 (3) [−] | 0.60 0.16 (2) [−] |
| | BicFeat | RF | 0.56 0.15 (5) [−] | 0.56 0.12 (4) [−] | 0.59 0.14 (3) [−] | 0.56 0.13 (4) [−] |
| Singh | None | SVM | 0.92 0.14 (1) [+] | 0.92 0.15 (1) [+] | 0.92 0.14 (1) [+] | 0.92 0.14 (1) [+] |
| | Noise | SVM | 0.74 0.16 (3) [+] | 0.75 0.15 (3) [+] | 0.81 0.17 (3) [+] | 0.81 0.15 (3) [+] |
| | Class. | SVM | 0.76 0.14 (2) [+] | 0.79 0.17 (2) [+] | 0.82 0.18 (2) [+] | 0.82 0.15 (2) [+] |
| | BicClass | RF | 0.73 0.13 (4) [+] | 0.73 0.13 (4) [+] | 0.74 0.15 (5) [+] | 0.77 0.14 (4) [+] |
| | BicExt | kNN | 0.72 0.14 (5) [+] | 0.67 0.13 (5) [+] | 0.75 0.15 (4) [+] | 0.77 0.15 (4) [+] |
| | BicFeat | NB | 0.59 0.11 (6) [+] | 0.57 0.14 (6) [+] | 0.60 0.15 (6) [+] | 0.60 0.20 (5) [+] |

**Table 6**
Results with noise rate of 40%. Cases in which the statistical test indicated that the best BicNoise variant has significantly prevailed or been outperformed by the best non-BicNoise method are highlighted in bold.

| Dataset | | Alg. | NCAR | NAR(a) | NAR(b) | NNAR |
|---|---|---|---|---|---|---|
| WDBC | None | SVM | 0.98 0.06 (1) [+] | 0.98 0.06 (1) [+] | 0.98 0.06 (1) [+] | 0.98 0.06 (1) [+] |
| | Noise | SVM | **0.80 0.06** (**4**) [−] | **0.94 0.07** (**2**) [−] | **0.77 0.07** (**4**) [−] | **0.77 0.07** (**3**) [−] |
| | Class. | SVM | **0.79 0.07** (**5**) [−] | **0.94 0.05** (**2**) [−] | **0.77 0.06** (**4**) [−] | **0.77 0.06** (**3**) [−] |
| | BicClass | RF | **0.85 0.06** (**3**) [−] | **0.88 0.06** (**3**) [−] | **0.85 0.06** (**2**) [−] | **0.83 0.06** (**2**) [−] |
| | BicExt | RF | **0.85 0.07** (**3**) [−] | **0.87 0.06** (**4**) [−] | **0.85 0.06** (**2**) [−] | **0.83 0.07** (**2**) [−] |
| | BicFeat | NB | **0.88 0.06** (**2**) [+] | **0.88 0.08** (**3**) [+] | **0.84 0.07** (**3**) [+] | **0.83 0.09** (**2**) [+] |
| Park. | None | kNN | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] |
| | Noise | RF | 0.77 0.18 (2) [−] | 0.72 0.18 (5) [−] | 0.75 0.19 (3) [−] | 0.75 0.20 (2) [−] |
| | Class. | Tree | 0.77 0.17 (2) [−] | 0.73 0.20 (4) [−] | 0.75 0.18 (3) [−] | 0.75 0.18 (2) [−] |
| | BicClass | SVM | 0.69 0.15 (4) [−] | 0.73 0.18 (4) [−] | 0.75 0.19 (3) [−] | 0.74 0.18 (3) [−] |
| | BicExt | NB | 0.72 0.18 (3) [−] | 0.77 0.18 (3) [−] | 0.76 0.18 (2) [−] | 0.74 0.20 (3) [−] |
| | BicFeat | NB | 0.72 0.15 (3) [+] | 0.79 0.19 (2) [+] | 0.74 0.18 (4) [+] | 0.75 0.18 (2) [+] |
| Park. (2) | None | kNN | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] |
| | Noise | NB | 0.71 0.20 (5) [−] | 0.74 0.20 (4) [−] | 0.72 0.26 (3) [−] | 0.72 0.22 (2) [−] |
| | Class. | kNN | 0.72 0.16 (4) [−] | 0.73 0.20 (5) [−] | 0.72 0.24 (3) [−] | 0.72 0.22 (2) [−] |
| | BicClass | NB | 0.73 0.14 (3) [−] | 0.82 0.19 (2) [−] | 0.73 0.22 (2) [−] | 0.71 0.23 (3) [−] |
| | BicExt | NB | 0.75 0.17 (2) [−] | 0.82 0.23 (2) [−] | 0.73 0.20 (2) [−] | 0.71 0.23 (3) [−] |
| | BicFeat | NB | 0.75 0.14 (2) [−] | 0.80 0.16 (3) [−] | 0.69 0.19 (4) [−] | 0.69 0.19 (4) [−] |
| Sonar | None | Ada | 0.76 0.18 (1) [+] | 0.76 0.18 (1) [+] | 0.76 0.18 (1) [+] | 0.76 0.18 (1) [+] |
| | Noise | RF | 0.63 0.18 (2) [−] | 0.61 0.16 (2) [−] | 0.55 0.18 (4) [−] | 0.55 0.18 (4) [−] |
| | Class. | kNN | 0.63 0.18 (2) [−] | 0.61 0.16 (2) [−] | 0.56 0.18 (3) [−] | 0.56 0.18 (3) [−] |
| | BicClass | SVM | 0.63 0.16 (2) [−] | 0.59 0.16 (4) [−] | 0.55 0.17 (4) [−] | 0.55 0.17 (4) [−] |
| | BicExt | SVM | 0.63 0.16 (2) [−] | 0.60 0.17 (3) [−] | 0.57 0.16 (2) [−] | 0.57 0.19 (2) [−] |
| | BicFeat | NB | 0.57 0.10 (3) [+] | 0.58 0.11 (5) [+] | 0.56 0.15 (3) [+] | 0.54 0.14 (5) [+] |
| Singh | None | SVM | 0.92 0.14 (1) [+] | 0.92 0.14 (1) [+] | 0.92 0.15 (1) [+] | 0.92 0.14 (1) [+] |
| | Noise | Ada | 0.72 0.16 (2) [+] | 0.76 0.15 (3) [+] | 0.72 0.16 (3) [+] | 0.72 0.16 (2) [+] |
| | Class. | SVM | 0.69 0.16 (3) [+] | 0.77 0.15 (2) [+] | 0.69 0.16 (4) [+] | 0.69 0.16 (5) [+] |
| | BicClass | Ada | 0.68 0.17 (4) [−] | 0.76 0.16 (3) [−] | 0.67 0.14 (5) [−] | 0.70 0.16 (4) [−] |
| | BicExt | Ada | 0.66 0.15 (5) [−] | 0.76 0.14 (3) [−] | 0.73 0.15 (2) [−] | 0.71 0.16 (3) [−] |
| | BicFeat | Tree | 0.57 0.14 (6) [+] | 0.55 0.13 (4) [+] | 0.59 0.17 (6) [+] | 0.60 0.17 (6) [+] |

**Table 7**
Results with noise rate of 50%. Cases in which the statistical test indicated that the best BicNoise variant has significantly prevailed or been outperformed by the best non-BicNoise method are highlighted in bold.

| Dataset | | Alg. | NCAR | NAR(a) | NAR(b) | NNAR |
|---|---|---|---|---|---|---|
| WDBC | None | SVM | 0.98 0.06 (1) [+] | 0.98 0.06 (1) [+] | 0.98 0.06 (1) [+] | 0.98 0.06 (1) [+] |
| | Noise | RF | **0.79 0.07** (**5**) [−] | **0.79 0.06** (**4**) [−] | **0.72 0.07** (**5**) [−] | **0.72 0.07** (**5**) [−] |
| | Class. | RF | **0.78 0.06** (**6**) [−] | **0.78 0.07** (**5**) [−] | **0.72 0.07** (**5**) [−] | **0.72 0.07** (**5**) [−] |
| | BicClass | NB | **0.85 0.06** (**3**) [−] | **0.86 0.06** (**3**) [−] | **0.82 0.07** (**4**) [−] | **0.82 0.07** (**3**) [−] |
| | BicExt | NB | **0.84 0.06** (**4**) [−] | **0.86 0.06** (**3**) [−] | **0.84 0.07** (**2**) [−] | **0.83 0.07** (**2**) [−] |
| | BicFeat | NB | **0.87 0.06** (**2**) [+] | **0.87 0.05** (**2**) [+] | **0.83 0.06** (**3**) [+] | **0.81 0.08** (**4**) [+] |
| Park. | None | kNN | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] |
| | Noise | kNN | 0.69 0.15 (4) [−] | 0.69 0.17 (4) [−] | 0.76 0.17 (3) [−] | 0.75 0.18 (3) [−] |
| | Class. | kNN | 0.69 0.18 (4) [−] | 0.69 0.17 (4) [−] | 0.75 0.17 (4) [−] | 0.75 0.17 (3) [−] |
| | BicClass | NB | 0.67 0.17 (5) [−] | 0.65 0.18 (5) [−] | 0.74 0.20 (5) [−] | 0.74 0.17 (4) [−] |
| | BicExt | NB | 0.74 0.17 (2) [−] | 0.71 0.20 (3) [−] | 0.76 0.20 (3) [−] | 0.76 0.19 (2) [−] |
| | BicFeat | NB | 0.71 0.17 (3) [+] | 0.73 0.20 (2) [+] | 0.77 0.20 (2) [+] | 0.73 0.16 (5) [+] |
| Park. (2) | None | kNN | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] | 0.87 0.21 (1) [+] |
| | Noise | RF | 0.68 0.17 (5) [−] | **0.63 0.15** (**6**) [−] | 0.69 0.22 (4) [−] | 0.69 0.22 (2) [−] |
| | Class. | RF | 0.68 0.17 (5) [−] | **0.64 0.17** (**5**) [−] | 0.69 0.23 (4) [−] | 0.69 0.21 (2) [−] |
| | BicClass | NB | 0.71 0.14 (4) [−] | **0.71 0.14** (**4**) [−] | 0.69 0.25 (4) [−] | 0.67 0.23 (3) [−] |
| | BicExt | NB | 0.74 0.15 (2) [−] | **0.74 0.17** (**2**) [−] | 0.72 0.25 (2) [−] | 0.67 0.22 (3) [−] |
| | BicFeat | NB | 0.72 0.19 (3) [−] | **0.73 0.14** (**3**) [−] | 0.70 0.17 (3) [−] | 0.66 0.18 (4) [−] |
| Sonar | None | Ada | 0.76 0.18 (1) [+] | 0.76 0.18 (1) [+] | 0.76 0.18 (1) [+] | 0.76 0.18 (1) [+] |
| | Noise | kNN | 0.57 0.16 (3) [−] | 0.66 0.20 (2) [−] | 0.52 0.14 (4) [−] | 0.52 0.14 (3) [−] |
| | Class. | kNN | 0.61 0.15 (2) [−] | 0.66 0.18 (2) [−] | 0.52 0.14 (4) [−] | 0.52 0.14 (3) [−] |
| | BicClass | Ada | 0.61 0.14 (2) [−] | 0.63 0.18 (3) [−] | 0.53 0.17 (3) [−] | 0.52 0.16 (3) [−] |
| | BicExt | Ada | 0.61 0.14 (2) [−] | 0.63 0.21 (3) [−] | 0.54 0.17 (2) [−] | 0.52 0.16 (3) [−] |
| | BicFeat | Tree | 0.54 0.11 (4) [+] | 0.56 0.13 (4) [+] | 0.53 0.08 (3) [+] | 0.54 0.10 (2) [+] |
| Singh | None | SVM | 0.92 0.14 (1) [+] | 0.92 0.14 (1) [+] | 0.92 0.14 (1) [+] | 0.92 0.14 (1) [+] |
| | Noise | kNN | 0.67 0.20 (5) [−] | 0.68 0.14 (2) [−] | 0.67 0.15 (2) [−] | 0.67 0.15 (2) [−] |
| | Class. | kNN | 0.71 0.18 (2) [−] | 0.68 0.14 (2) [−] | 0.67 0.14 (2) [−] | 0.67 0.14 (2) [−] |
| | BicClass | Tree | 0.68 0.17 (4) [−] | 0.66 0.15 (4) [−] | 0.67 0.14 (2) [−] | 0.66 0.14 (3) [−] |
| | BicExt | Tree | 0.69 0.18 (3) [−] | 0.67 0.14 (3) [−] | 0.67 0.13 (2) [−] | 0.66 0.15 (3) [−] |
| | BicFeat | NB | 0.56 0.11 (6) [+] | 0.58 0.15 (5) [+] | 0.57 0.13 (3) [+] | 0.54 0.13 (4) [+] |

**Table 8**
Summary of performance results – aggregated rankings via the Borda count method (Langville & Meyer, 2012).

| Noise | Alg. | 10% | 20% | 30% | 40% | 50% |
|---|---|---|---|---|---|---|
| NCAR | None | 30 (1) | 30 (1) | 30 (1) | 30 (1) | 30 (1) |
| | Noise | 26 (3) | 22 (3) | 23 (3) | 20 (2) | 13 (5) |
| | Class. | 27 (2) | 23 (2) | 24 (2) | 19 (3) | 16 (4) |
| | BicClass | 19 (5) | 18 (4) | 16 (5) | 19 (3) | 17 (3) |
| | BicExt | 21 (4) | 17 (5) | 18 (4) | 20 (2) | 22 (2) |
| | BicFeat | 18 (6) | 10 (6) | 12 (6) | 19 (3) | 17 (3) |
| NAR(a) | None | 30 (1) | 30 (1) | 30 (1) | 30 (1) | 30 (1) |
| | Noise | 24 (3) | 19 (4) | 21 (2) | 19 (3) | 17 (4) |
| | Class. | 25 (2) | 21 (2) | 21 (2) | 20 (2) | 17 (4) |
| | BicClass | 18 (5) | 20 (3) | 18 (3) | 19 (3) | 16 (5) |
| | BicExt | 20 (4) | 21 (2) | 17 (4) | 20 (2) | 21 (2) |
| | BicFeat | 16 (6) | 13 (5) | 13 (5) | 18 (4) | 19 (3) |
| NAR(b) | None | 30 (1) | 30 (1) | 30 (1) | 30 (1) | 30 (1) |
| | Noise | 23 (3) | 19 (4) | 19 (3) | 18 (4) | 17 (4) |
| | Class. | 24 (2) | 21 (2) | 19 (3) | 18 (4) | 16 (5) |
| | BicClass | 23 (3) | 20 (3) | 19 (3) | 19 (3) | 17 (4) |
| | BicExt | 21 (4) | 20 (3) | 20 (2) | 25 (2) | 24 (2) |
| | BicFeat | 14 (5) | 15 (5) | 16 (4) | 15 (5) | 21 (3) |
| NNAR | None | 30 (1) | 30 (1) | 30 (1) | 30 (1) | 30 (1) |
| | Noise | 23 (2) | 19 (4) | 21 (3) | 22 (2) | 20 (3) |
| | Class. | 23 (2) | 19 (4) | 21 (3) | 20 (3) | 20 (3) |
| | BicClass | 19 (4) | 24 (2) | 22 (2) | 19 (4) | 19 (4) |
| | BicExt | 22 (3) | 20 (3) | 21 (3) | 22 (2) | 22 (2) |
| | BicFeat | 13 (5) | 17 (5) | 15 (4) | 16 (5) | 16 (5) |

of conditional independence among the attributes and the use of conditional probabilities to derive posterior probabilities, SVM and KNN has as a weak feature the reliance on each single instance to derive the decision model, so that the decision boundaries among classes can be easily altered by the inclusion or exclusion of a few noisy instances.

## 5. Final remarks

In this paper, we have introduced a novel biclustering approach, referred to as BicNoise, for coping with mislabeled data in binary classification problems. To the best of our knowledge, this is the first study fully investigating the potentials of biclustering as a feature extraction technique in the label noise context, which is the main theoretical contribution of the paper.

Three variants of BicNoise have been formally described and empirically assessed through a systematic series of experiments, which involved different rates and types of label noise, and also different types of classifiers, datasets, and evaluation metrics. Such thorough empirical study complements others available in the literature (Nettleton et al., 2010), and this can be regarded as another source of contributions. Overall, the results achieved evidence the usefulness of BicNoise in circumventing the mislabeling problem, mainly when high rates of noise are considered. So, this study has contributed with an approach that is not only conceptually novel but also practically effective.

The results most noticeably show that BicNoise is better suited in two situations: (i) whenever the dataset contains correlated attributes and; (ii) when the data has a tendency of having a higher noise level. This is due to the fact that the biclustering technique reduces the effect of the curse of dimensionality by selectively filtering uninformative attributes, thus helping the classifier with its duties. This is evidenced by noticing that mostly the classifier with the best results on the transformed data is the simple Naïve Bayes approach.

Regarding the different variants of BicNoise, BicExt and BicClass performed better than BicFeat, since BicFeat may oversimplify the data description. However, one positive aspect of BicFeat is that it usually leads to sparse binary datasets, which may entail gains in performance during the classifier training. BicExt, on the other hand, seems to be more robust under the presence of noise, since it does not rely on the class information, whereas BicClass was slightly better in lower noise settings. So, if one suspects that high levels of data mislabeling takes place, the results suggest that BicExt should be the preferred alternative.

Notice that since BicNoise transforms the dataset (either in an unsupervised or supervised fashion) it can complement other data cleansing or (supervised) noise-tolerant approaches as well. Also, instead of many techniques proposed to deal with noisy labels, the BicFeat and BicExt variants do not rely on the labeled data, thereby avoiding that a mislabeled instance around the decision boundary has any influence on the correction mechanisms.

As future work, we intend to enlarge the scope of investigation by also considering the multiclass classification scenario as well as other biclustering algorithms to perform the first step of our approach. In fact, BicNoise can be seen as a very flexible conceptual framework aiming to cope with noisy data through biclustering. So, the study of the applicability of other biclustering algorithms/models (Cheng & Church, 2000; Madeira & Oliveira, 2004; Dhillon, 2001; Dhillon et al., 2003; Prelic et al., 2006) in this context seems to be an interesting line of investigation to be pursued. We also intend to assess the performance of BicNoise variants when applied to highly imbalanced datasets as well as to high dimensional/small sample settings, which is typical in bioinformatics problems (de Souto et al., 2008). Additionaly, we shall investigate alternatives to perform feature selection/weighting on the feature sets produced by the BicNoise variants, taking as inspiration other approaches available in the literature (Frénay, Doquire, & Verleysen, 2014). Finally, we also plan to understand more thoroughly the class discrimination capabilities of the novel feature spaces generated by the induced bicluster set.

## Acknowledgments

This work has been financially supported by the National Council for Scientific and Technological Development (CNPq/Brazil), under Grant No. 304603/2012-0.

## Appendix A

Fig. 2 makes it possible to visually inspect the correlations between the performance metrics adopted in our experiments, as measured pairwise via the Pearson correlation coefficient. In this figure, the black (red) squares mean that a given measure among Accuracy, Precision, Recall, and F-measure is (not) strongly correlated to AUC, according to the application of a statistical test. Each square of a 4 × 4 submatrix denotes a particular experimental setting, that is, a particular combination of dataset, noise rate, and noise type (each row in a submatrix corresponds to one of the types shown in the third column of Table 1). The correlations were calculated over the results produced by all combinations of classifiers and denoising strategy.

As one can notice, in all but one case, Accuracy is strongly correlated with AUC. Moreover, the correlation between F-measure and AUC is mostly affected only for 40% as noise rate, being restored afterwards. On the other hand, sometimes, either Precision or Recall do not correlate strongly with AUC, but this does not happen at the same time for both measures. So, when one measure does not correlate strongly, the other does in a way that preserves the high correlation of F-measure in most of the times. Finally, it is interesting to notice that the noise type has not much effect on the preservation of the correlations (usually, the squares of a given submatrix column are all back or red), while the same observation cannot be made with respect to the effect due to the classification problem (dataset).

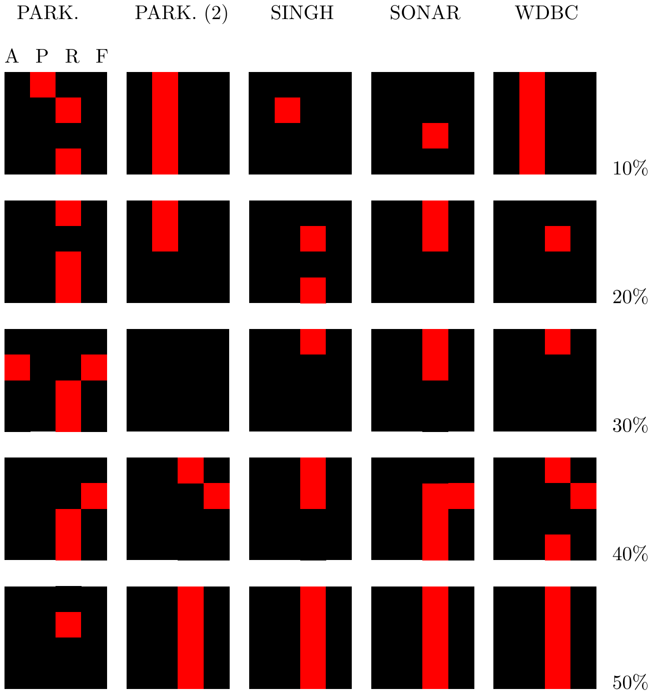

**Fig. 2.** Correlation between AUC and the other metrics, as measured via Pearson correlation, considering different configuration settings. Here, 'A', 'P', 'R', and 'F' stand for Accuracy, Precision, Recall, and F-measure, respectively.

## References

Abellán, J., & Masegosa, A. R. (2012). Bagging schemes on the presence of class noise in classification. *Expert Systems with Applications, 39*, 6827–6837.

Abellán, J., & Moral, S. (2003). Building classification trees using the total uncertainty criterion. *International Journal of Intelligent Systems, 18*, 1215–1225.

Agrawal, R., Gehrke, J., Gunopulus, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In: *Proceedings of the ACM/SIGMOD int. conference on management of data* (pp. 94–105).

Allison, P. D. (2002). Missing data: Quantitative applications in the social sciences. *British Journal of Mathematical and Statistical Psychology, 55*, 193–196.

Atla, A., Tada, R., Sheng, V., & Singireddy, N. (2011). Sensitivity of different machine learning algorithms to noise. *Journal of Computing Sciences in Colleges, 26*, 96–103.

Bootkrajang, J., & Kabán, A. (2012). Label-noise robust logistic regression and its applications. In *Machine learning and knowledge discovery in databases – European conferences ECML PKDD 2012* (pp. 143–158).

Bootkrajang, J., & Kabán, A. (2014). Learning kernel logistic regression in the presence of class label noise. *Pattern Recognition, 47*, 3641–3655.

Bouveyron, C., & Girard, S. (2009). Robust supervised classification with mixture models: Learning from data with uncertain labels. *Pattern Recognition, 42*, 2649–2658.

Brodley, C. E., & Friedl, M. A. (1999). Identifying mislabeled training data. *Journal of Artificial Intelligence Research, 11*, 131–167.

Cantador, I., & Dorronsoro, J. R. (2005). Boosting parallel perceptrons for label noise reduction in classification problems. *Proceedings of the first international work-conference on the interplay between natural and artificial computation conference on artificial intelligence and knowledge engineering applications: A bioinspired approach (IWINAC'05)* (Vol. Part II, pp. 586–593). Springer.

Cao, J., Kwong, S., & Wang, R. (2012). A noise-detection based adaboost algorithm for mislabeled data. *Pattern Recognition, 45*, 4451–4465.

Cheng, Y., & Church, G. M., 2000. Biclustering of expression data. In *Proceedings of the eighth int. conf. on intelligent systems for molecular biology* (pp. 93–103).

Coates, A., Ng, A. Y., & Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of international conference on artificial intelligence and statistics* (pp. 215–223).

Coelho, G. P., de França, F. O., & Von Zuben, F. J. (2009). Multi-objective biclustering: When non-dominated solutions are not enough. *Journal of Mathematical Modelling and Algorithms, 8*, 175–202.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory*. Wiley.

de Castro, P. A. D., de França, F. O., Ferreira, H. M., & Von Zuben, F. J. (2007). Applying biclustering to text mining: An immune-inspired approach. In de Castro, L. N., Von Zuben, F. J., Knidel, H. (Eds.), *Artificial immune systems, proceedings of the sixth international conference on artificial immune systems (ICARIS), Santos, Brazil* (pp. 83–94).

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30.

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh int. con. on knowledge discovery and data mining* (pp. 269–274).

Dhillon, I. S., Mallela, D. S., & Modha, S. (2003). Information theoretic co-clustering. In *Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining* (pp. 89–98). Springer.

Dorigo, M., Bonabeau, E., Theraulaz, G., et al. (2000). Ant algorithms and stigmergy. *Future Generation Computer Systems, 16*, 851–871.

Duch, W., Jankowski, N., & Maszczyk, T. (2012). Make it cheap: Learning with $O(nd)$ complexity. In *Proceedings of 2012 IEEE international joint conference on neural networks (IJCNN)* (pp. 1–4).

de França, F., & Von Zuben, F. (2010). Finding a high coverage set of $\delta$-biclusters with swarm intelligence. In *Proceedings of 2010 IEEE congress on evolutionary computation (CEC)* (pp. 1–8).

de França, F., & Von Zuben, F. J. (2011). Extracting additive and multiplicative coherent biclusters with swarm intelligence. In: *Proceedings of 2011 IEEE congress on evolutionary computation (CEC)* (pp. 632–638).

de França, F. O. (2012). Scalable overlapping co-clustering of word-document data. In *Proceedings of 2012 11th international conference on machine learning and applications (ICMLA)* (pp. 464–467).

de França, F. O., Coelho, G. P., & Zuben, F. J. V. (2013). Predicting missing values with biclustering: A coherence-based approach. *Pattern Recognition, 16*, 1255–1266.

Frénay, B., Doquire, G., & Verleysen, M. (2014). Estimating mutual information for feature selection in the presence of label noise. *Computational Statistics & Data Analysis, 71*, 832–848.

Frénay, B., & Verleysen, M. (2014). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems, 25*, 845–869.

Guan, D., Yuan, W., Lee, Y. K., & Lee, S. (2011). Identifying mislabeled training data with the aid of unlabeled data. *Applied Intelligence, 35*, 345–358.

Guan, D., Yuan, W., Ma, T., & Lee, S. (2014). Detecting potential labeling errors for bioinformatics by multiple voting. *Knowledge-Based Systems, 66*, 28–35.

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computing, 18*, 1527–1554.

Langville, A. N., & Meyer, C. D. (2012). *Who's the #1? The science of rating and ranking*. Princeton University Press.

Little, R. J. A., & Rubin, D. B. (2002). *Statistical analysis with missing data*. Wiley.

Madeira, S., & Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1*, 24–45.

Mantas, C. J., & Abellán, J. (2014a). Analysis and extension of decision trees based on imprecise probabilities: Application on noisy data. *Expert Systems with Applications, 41*, 2514–2525.

Mantas, C. J., & Abellán, J. (2014b). Credal-c4. 5: Decision tree based on imprecise probabilities to classify noisy data. *Expert Systems with Applications, 41*, 4625–4637.

Nettleton, D. F., Orriols-Puig, A., & Fornells, A. (2010). A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review, 33*, 275–306.

Prati, R. C., & de França, F. O. (2013). Extending features for multilabel classification with swarm biclustering. In *Proceedings of 2013 IEEE congress on evolutionary computation (CEC)* (pp. 2964–2971). IEEE.

Prelic, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., et al. (2006). A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics, 22*, 1122–1129.

Rebbapragada, U., & Brodley, C. E. (2007). Class noise mitigation through instance weighting. In *Proceedings of the 18th European conference on machine learning (ECML)* (pp. 708–715). Springer.

Rider, A. K., Johnson, R. A., Davis, D. A., Hoens, T. R., & Chawla, N. V. (2013). Classifier evaluation with missing negative class labels. In *Advances in intelligent data analysis XII – 12th international symposium, IDA 2013* (pp. 380–391). Springer.

de Souto, M. C. P., Costa, I. G., de Araujo, D. S. A., Ludermir, T. B., & Schliep, A. (2008). Clustering cancer gene expression data: A comparative study. *BMC Bioinformatics, 9*, 497.

Symeonidis, P., Nanopoulos, A., Papadopoulos, A., & Manolopoulos, Y. (2008). Nearest-biclusters collaborative filtering based on constant and coherent values. *Information Retrieval, 11*, 51–75.

Tabassian, M., Ghaderi, R., & Ebrahimpour, R. (2012a). Combination of multiple diverse classifiers using belief functions for handling data with imperfect labels. *Expert Systems with Applications, 39*, 1698–1707.

Tabassian, M., Ghaderi, R., & Ebrahimpour, R. (2012b). Combining complementary information sources in the Dempster–Shafer framework for solving classification problems with imperfect labels. *Knowledge-Based Systems, 27*, 92–102.

Wang, X., Liu, F., Jiao, L. C., Zhou, Z., Yu, J., Li, B., et al. (2012). An evidential reasoning based classification algorithm and its application for face recognition with class noise. *Pattern Recognition, 45*, 4117–4128.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques*. Elsevier.

Wu, X., & Zhu, X. (2008). Mining with noise knowledge: Error-aware data mining. *IEEE Transactions on Systems, Man, and Cybernetics, Part A, 38*, 917–932.

Zhu, X., & Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review, 22*, 177–210.

Zhu, X., Wu, X., & Chen, Q. (2006). Bridging local and global data cleansing: Identifying class noise in large, distributed data datasets. *Data Mining and Knowledge Discovery, 12*, 275–308.

Zhu, X., Wu, X., Khoshgoftaar, T. M., & Shi, Y. (2007). An empirical study of the noise impact on cost-sensitive learning. In *Proceedings of international joint conference on artificial intelligence (IJCAI)* (pp. 1168–1173). Morgan Kaufmann.