Knowledge and Information Systems (2019) 60:63–97
https://doi.org/10.1007/s10115-018-1244-4

REGULAR PAPER

Emerging topics and challenges of learning from noisy data in nonstandard classification: a survey beyond binary class noise

Ronaldo C. Prati[1] · **Julián Luengo**[2] · **Francisco Herrera**[2]

Received: 17 February 2017 / Revised: 16 May 2018 / Accepted: 16 June 2018 /
Published online: 6 July 2018

Abstract
The problem of class noisy instances is omnipresent in different classification problems. However, most of research focuses on noise handling in binary classification problems and adaptations to multiclass learning. This paper aims to contextualize noise labels in the context of non-binary classification problems, including multiclass, multilabel, multitask, multi-instance ordinal and data stream classification. Practical considerations for analyzing noise under these classification problems, as well as trends, open-ended problems and future research directions are analyzed. We believe this paper could help expand research on class noise handling and help practitioners to better identify the particular aspects of noise in challenging classification scenarios.

Keywords Class noise · Multiclass · Multilabel · Multitask · Multi-instance · Ordinal classification · Data streams

1 Introduction

Learning from noisy data is an important topic in machine learning, data mining and pattern recognition, as real-world data sets may suffer from imperfections in data acquisition, transmission, storage, integration and categorization. Indeed, over the last few decades, noisy data have attracted a considerable amount of attention from researchers and practitioners, and the research community has developed numerous techniques and algorithms in order to deal with the issue [30,41,147].

These approaches include the development of learning algorithms which are robust to noise as well as data preprocessing techniques that remove or "repair" noisy instances. Although noise can affect both input and class attributes, class noise is generally considered more

✉ Ronaldo C. Prati
ronaldo.prati@ufabc.edu.br

[1] Center of Mathematics, Computer Science and Cognition (CMCC), Federal University of ABC (UFABC), Santo André, São Paulo, Brazil

[2] Department of Computer Science and A.I. (DECSAI), University of Granada (UGR), Granada, Spain

harmful to the learning process, and methods for dealing with class noise are becoming more frequent in the literature [147].

Class noise may have many reasons, such as errors or subjectivity in the data labeling process, as well as the use of inadequate information for labeling. For instance, in some medical applications, the true status of some diseases can only be determined by expensive or intrusive procedures, some of which can only be carried out after a patient's death. Another reason is that data labeling by domain experts is generally costly, and several applications use labels which are automatically defined by autonomous taggers (e.g., sentiment analysis polarization [70]), or by non-domain experts. This approach is common in, e.g., social media analysis [70], where hashtags used by users or information provided by a pool of non-domain experts (crowdsourcing) are used to derive labels.

Even though class noise is predominant in the literature (see Fréenay and Verleysen [30], Nettleton et al. [90] for recent surveys and comparison studies), most of the research has been focused on noise handling in binary class problems. However, new real-life problems have motivated the development of classification paradigms beyond binary classification [50]. These paradigms include ordinal class [48], multiclass [25], multilabel [51] and multi-instance [52] as well as learning from data streams and non-stationary environments [26] and joint exploiting related tasks [28]. Due to the ubiquity of noise, it is of fundamental importance to better understand the relationships and implications of class noise within these paradigms. Each paradigm has its own particularities which impose new challenges and research questions for noise handling. Although research for class noise handling in these paradigms is somewhat present in the literature (as will be discussed further in this paper), it remains quite scarce and requires general discussion of issues, challenges and research practices regarding it. This paper aims to discuss open-ended challenges and future research directions for learning with class noise data, focusing on non-binary classification problems. The main contributions of this paper are:

- We discuss some current research, as well as the need of adaptation or development of new techniques for handling class noise within non-binary classification paradigms.
- We also discuss issues related to the simulation of noise scenarios (inclusion of artificial noise) within these paradigms, an experimental artifact frequently adopted for analysis of noise dealing techniques. These issues are important for simulating noise scenarios that may occur in real-world applications and can serve as the basis for uniforming procedures by providing an objective ground in order to assess the robustness of the learning methods.
- We present some important open-ended issues and offer some possible solutions to the existing problems.

We believe this discussion will encourage researchers and practitioners to explore the problem of class noise handling in new scenarios and different learning paradigms in more detail. The rest of this paper is organized as follows: Sect. 2 presents an overview of techniques and methods for learning from class noise. Section 3 presents a discussion of class noise in the context of six non-binary classification problem: multiclass, multilabel , multitask, multi-instance, ordinal and data streams classification. Section 4 is devoted to present the latest advances in each non-binary classification problem and their respective open issues. Finally, Sect. 5 presents some concluding remarks.

2 An overview of learning with noise

The development of supervised learning algorithms and methods that can handle data sets with class noise is a problem of great practical importance, as numerous real-world problems are noisy [89]. Class noise may have many reasons [30], such as insufficient or imperfect information [11]; errors in data labeling due to the subjectivity [133], the use of automated processes [119] or non-domain experts in data labeling [55]; or encoding or communication problems [12]. In this section, we introduce the basics of label noise: Sect. 2.1 briefly introduces the problem of learning with class noise. Section 2.2 presents a class noise taxonomy based on statistical mechanism that introduce noise; Sect. 2.3 discuss some potential impacts of noisy labels in the learning algorithms, and Sect. 2.4 is devoted to the different techniques developed to cope with class noise.

2.1 Problem statement

A training data set D is composed of a collection of examples (X, Y). Examples are pairs $(\mathbf{x_i}, \mathbf{y_i})$, where $\mathbf{x} \in X$ is described by attributes from the input space $\mathcal{X}$, and their corresponding targets $\mathbf{y_i} \in Y$ from the attributes of the output space $\mathcal{Y}$, associated with the underline and unknown function $\mathcal{F} : \mathcal{X} \rightarrow Y$. A learning algorithm aims to create, from D, a model function $\mathcal{H} : \mathcal{X} \rightarrow \mathcal{Y} \approx \mathcal{F}$, that can be applied to predict the target class variables of new instances. In this paper, Y can be formed by a single-class attribute, as in binary or multiclass classification cases, or by more than a single-class attribute, as in multilabel classification cases.

Noise may corrupt the values of attributes in X and Y. In this paper, we restrict the discussion in the target (class) attributes, as class noise is known to be more harmful to the learning process [147]. In this case, instead of the true target values Y, the learner has access to the corrupted values $\hat{Y}$. The objective of noise handling methods is to learn a good approximation of $\mathcal{H}' : \mathcal{X} \rightarrow \mathcal{Y} \approx \mathcal{F}$ from the noise data set $D' = (X, \hat{Y})$. To this end, the side effects of noisy instances (when $Y \neq \hat{Y}$) in the learning process should be diminished. Methods for dealing with class noisy data include data- and algorithm-level approaches. In this paper, we are interested in noise introduced by stochastic processes, rather than noise introduced deliberately by an adversarial agent [130]. A noise model aims to describe some of the statistical properties of how noise appears in a data set.

2.2 Class noise models

In Fréenay and Verleysen [30], a taxonomy of class noise mechanisms was proposed. This taxonomy is inspired by the missing value taxonomy [69], adapted to the class noise context. The taxonomy is based on four random variables: X is the feature vector, Y is the true class, $\hat{Y}$ is the observed class, and E is a binary variable that indicates whether noise is present, that is, $p_e = P(E = 1) = P(Y \neq \hat{Y})$. This taxonomy is only applicable to binary and multiclass problems.

According to the statistical dependencies among these four variables, three different statistical models arise. They are briefly represented in Fig. 1, where arrows are used to depict statistical dependence among the variables. In this taxonomy, the class noise occurrence is assumed to be a stochastic process and the probability p_e of an instance being mislabeled is categorized into three groups:

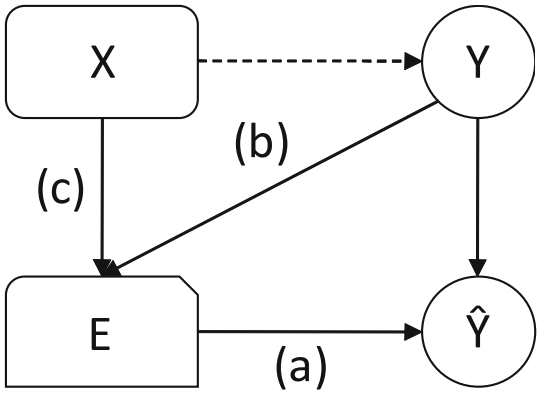

Fig. 1 Simplified statistical dependence model from [30] summarizing the class noise typologies. The arrows indicate statistical dependence. Marked arrows (a), (b) and (c) are relevant for the three statistical models considered: noise completely at random (NCAR) (a); noise at random (NAR) (a) and (b) and noise not at random (NNAR) (a), (b) and (c). The dashed line represents the implicit dependence among the input variables and the class labels

Noise Completely at Random (NCAR) This type of noise occurs in a completely stochastic way, and the probability of an instance being mislabeled does not depend on the class nor the other predictive features. Please note that in binary classification problems, NCAR becomes symmetrical since an asymptotically identical proportion of instances can be corrupted for both classes.

Noise at Random (NAR) In this type of noise, the probability of a mislabeling is dependent on the value of the actual class, i.e., it can assume different values for different classes. (Some classes might be noisier than others.) However, the probability of an instance being mislabeled is the same for all instances within each class. NCAR class noise is a particular case of NAR class noise, where the probabilities are the same for all classes.

Noise not at Random (NNAR) Both NCAR and NAR models assume that class noise applies to all instances, but this is not always true. In NNAR, the probability of an instance be mislabeled depends somehow on the feature space: for example, instance near class boundaries, similar to instances of another class label, are likely to be noisier.

This taxonomy is sufficient for binary classification problems (as it is based on missing value case, where a value is either missing or not) but as is argued in Sect. 3, non-binary and nonstandard classifications problems may require additional dependencies. To the best of our knowledge, such dependencies are not properly discussed in the literature.

2.3 Impact of class noise in learning algorithms

Class noise may have significant impacts on the learning process. The deterioration in classification performance is perhaps the most frequently studied problem in the literature, from both theoretical and experimental perspectives.

Theoretical studies generally depart from some assumptions of data distribution (e.g., data came from a multivariate Gaussian distribution with identical covariance matrix [81] or unequal covariance matrices [61,83]), combined with models with known mathematical properties (e.g., a linear discriminant function) to establish bounds on errors in the presence of class noise.

The impact of class noise for some specific algorithms has also been addressed theoretically in the literature. Bi and Jeske [8] study the behavior of logistic regression in comparison with discriminant analysis, under NCAR noise assumption, concluding that logistic regression is less affected. In Okamoto and Yugami [93], average class analysis was used to

investigate the behavior of class noise in instance-based learning algorithms. The impact of class noise on decision tree induction was studied in Quinlan [103].

The PAC-learning framework was also used to theoretically study the effect of class noise in classification tasks. In Angluin and Laird [3], the NCAR noise model is assumed and PAC learning was used to estimate the noise rate in a data set, as long as less than 50% of instances were noisy. In recent studies [56,107], PAC learning is used to provide some insights into conditions in which learning under different class noise models (and not only NCAR) can be successful.

From a empirical perspective, studies dealing with noisy instances are often performed using simulation, by artificially introducing noise according to some noise assumptions [90]. The reasons are twofold: first, few data sets with known noisy instances are available; and second, simulated experiments allow the control of parameters and characteristics of noise introduction in the data set. However, the way noise is artificially introduced may follow different procedures, which may influence the behavior of methods for dealing with noise data. To alleviate this problem, it is important to uniform artificial noise inclusion nomenclatures, as in the taxonomy discussed in the previous section. This paper extends this taxonomy by including nonstandard classification problems, as well as more detailed noise models for multiclass classification.

Some studies reported in the literature have also shown the impact of class noise in feature selection [143], an increase in model complexity [1], and the distortion of observed frequencies [32], among others.

2.4 Strategies for coping with noise

There are traditionally two different strategies for dealing with class noise, each one related to the phase they operate:

Algorithm-level approaches Their aim is to design robust classifiers. They are less influenced by noise and do not require any previous noise treatment [82,120].
Data-level approaches They attempt to remove or cleanse the noise present in the data before applying a classifier [12,37]. This is a popular option if using a robust learner is unfeasible or inappropriate, or even aiming to improve results of robust learners, enabling the selection of any standard classification algorithm.

Recent studies have enriched the taxonomy presented above. For instance, algorithm-level approaches can be differentiated between approaches that either model the presence of noise explicitly or implicitly. Data-level approaches can also comprise filtering or reparation methods if analyzed in detail. Since the categorization of noise treatment methods is richer than the traditional literature categorization, methods for dealing with noise were divided into five categories:

Data Cleaning Methods A possible approach for dealing with noise consists in somehow cleaning the noisy instances in the data set by removing them or repairing their labels. However, identifying class noisy instances is not an easy task, and several approaches have been proposed. These approaches include using data characteristic measures [38], building predictive models for identification [150], cost-sensitive models [148], ensembles [113], nearest neighborhood [60], influence of the instance in model creation [141] and graph-based methods [115], to name a few.
Noise Reparation This category includes those techniques that aim to correct or repair noisy instances by changing the noisy label fir the correct estimated one [124]. We must

differentiate between pure relabeling (or correcting) methods, which only change the noisy instance label [91,140] and complete noise cleaners that are able to both remove and relabel the noisy instances [86]. As some authors point out [147], noise reparation in training data, even when the test data is also noisy, will enhance the model's performance compared to doing nothing about noise.

Noise Robust Methods These methods are able to learn useful methods without noise modeling or data cleaning even when some amounts of noise labeling are present. Theoretical studies which aim to investigate noise robustness [75] or different loss functions were studied for the binary classification problem [5]. Overfitting avoidance techniques, such as noise robust splitting criteria in decision trees [104] and regularization [57], are generally used to increase noise robustness. Another common practice to increase noise robustness is the use of ensembles to increase the robustness of the base classifiers [113,117] or to detect noisy instances in the building process [111]. Even popular disciplines such as deep learning evaluate their robustness against label noise, which is crucial in real-world applications [110].

Probabilistic Noise Tolerant Methods Some approaches for dealing with noise use a probabilistic noise modeling technique to simultaneously incorporate a noise model in model construction. This is designed to model the (unknown) noise-free distribution by decoupling the data and noise generation process. Some examples include Bayesian approaches, by incorporating the prior probability distribution of noise [27]; subpopulation approaches by using a mixture model of normal and noise distributions [116]; and belief functions to model the degree of certainty in a class label [73].

Model-based Noise Tolerant Methods Some artifacts can be incorporated in a learning algorithm in order to increase tolerance to noise. For instance SVMs [44,65,137], neural networks [58] and ensembles [99] can incorporate mechanisms for noise tolerance.

In Fig. 2, we depict a combined taxonomy derived from Fréenay and Verleysen [30] related to the traditional approaches described in the literature. As the reader may notice, the *Data Cleaning Methods* category is directly related to the *data-level approaches*, while we have opted to distinguish between pure noise filtering methods and repairing algorithms that relabel the identified noisy instances.

The reader must take into account that, when dealing with noisy data, the statistical model which induced the noise should be identified. Failing to do so will result in severe bias in the resulting preprocessed data or the created model. From the knowledge of missing values literature [69], filtering instances is acceptable when NAR or NCAR noisiness models apply. If NNAR noise is detected, more complex strategies should be used: since the noise introduction is dependent on the input attributes, the latter cannot be safely used to identify mislabeled instances.

From a statistical point of view, a classifier estimates the probability $P(Y|X)$. Following the dependences depicted in Fig. 1, the NAR case will result in $P(\hat{Y}|E, Y)$, which is independent in terms of X on the estimation made by the classifier $P(Y|X)$. However, in the NNAR case we have $P(\hat{Y}|X, E, Y)$, which is not statistically independent of the distribution of X. As a result, the noise will bias a traditional classifier, whose assumption is such that $P(X|Y)$ is the only dependence among the class label and the observed (input) values.

The NNAR framework implies that the classifier must be aware of the dependence between the input values and the observed (and possibly not true) class label [75]. In these cases, the usage of probabilistic noise tolerant methods is recommended, as they enable the practitioner to model such dependence on both the input values and the observed class. Other approaches

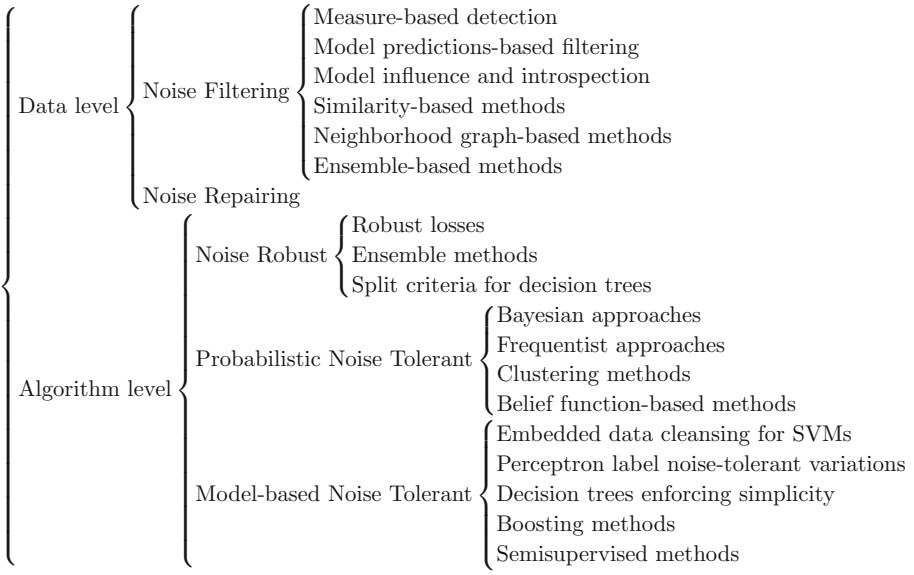

Fig. 2 Taxonomy of noise treatment methods, adapted from [30] to meet previous nomenclature

can be used, as in risk minimization, where different loss functions have been analyzed under the NNAR assumption [6].

3 Noise characterization in non-binary classification problems

While learning from noisy data is a challenge in itself, open-ended problems in machine learning would make learning with noisy data more difficult to handle. This section analyzes the class noise problem under the context of non-binary classification problems, including multiclass, multilabel, multitask, multi-instance, ordinal and data streams classifications. Each problem presents characteristics which require specialized noise handling techniques, as well as particularities that must be addressed when designing experimental evaluations. Even some well-studied problems such as noise handling in multiclass classification have some peculiarities which have not been properly discussed in the literature.

3.1 Multiclass noise classification

In multiclass classification, each instance is assigned to one class of the set $\mathcal{Y} = \{c_1, \ldots, c_m\}$, where $m > 2$ is the cardinality of $\mathcal{Y}$, i.e., the number of possible classes. Class noise in this case occurs when a true label for some arbitrary class c_i is erroneously replaced by another label in $\mathcal{Y} \backslash \{c_i\}$ subset.

A lager number of classes are a source of additional trade-offs for machine learning algorithms [4]. A similar phenomenon happens in noise modeling. As seen in Sect. 2, traditional statistical taxonomy of label noise considers three categories, which takes into account

whether the occurrence of a class noise depends or not on the true class, or the input attribute values. Although this taxonomy is still applicable to multiclass problems, the multiclass scenario introduces another stochastic dependencies on how the other classes influence the definition of the observed noisy class.

In those cases where noise does not depend on data features (NCAR and NAR), the noise process can be categorized into terms of the labeling or transition matrix [30,98], as shown in Eq. 1.

$$y = \begin{pmatrix} \gamma_{11} & \cdots & \gamma_{1m} \\ \vdots & \ddots & \dots \\ \gamma_{m1} & \cdots & \gamma_{mm} \end{pmatrix} \tag{1}$$

where γ_{ij}, for $i, j = \{1 \dots m\}$, is the probability of an instance whose true label is y_i which has been (mis)labeled as y_j. Each row in the labeling matrix sums up to 1, as each instance should be assigned to one label.

In the NCAR case, the probability p_e of an instance being mislabeled does not depend on the class nor the input space. Therefore, the probability γ_{ii} of an instance not being mislabeled is the same for all classes (as it does not depend on the class values) and is equal to $1 - p_e$. Thus, the main diagonal in the labeling matrix has the same value.

The NAR case allows asymmetrical mislabeling for each class. Let $\mathcal{P} = \{p_1, \dots, p_m\}$, be the mislabeling rate associated with each class and $\Pi = \{\pi_1, \dots, \pi_m\}$ the class distribution for each class. The overall probability is proportional to the error probability of each class and its corresponding distribution, i.e., $p_e = \sum_i^m p_i \pi_i$. Thus, $\gamma_{ii} = 1 - p_i$. Note that NCAR is a special case of NAR, in which p_i is the same for all m classes.

Equation 2 shows the labeling matrix for binary class problems. Note that, as only two classes are available, the matrix is 2×2. Under NAR, $p_1 = p_2 = p_e$, and for both NAR and NCAR, the probabilities of λ_{ij} for $i \neq j$ are uniquely and unambiguously defined as the complement of λ_{ii}, as each row must sum up to one.

$$y_{\text{bin}} = \begin{pmatrix} 1 - p_1 & p_1 \\ p_2 & 1 - p_2 \end{pmatrix} \tag{2}$$

However, in problems with more than two classes, the distribution of noise probabilities is no longer uniquely defined, as they should be distributed among the remaining classes. Although there are numerous possibilities to consider, some of them quite arbitrary [62], and we have highlighted some general cases which may be of practical interest.

Uniform distribution The most common approach considered in the literature is the uniform case [41], where the observed noise is evenly distributed among the remaining classes. This is equivalent to saying that, when noise for a certain class occurs, all remaining classes are just as equally probable as the observed label of the instance. The corresponding labeling matrix is shown in Eq. 3. In the NCAR and NAR cases, the probabilities γ_{ij} (for $i \neq j$) are $p_e/m-1$ (as $p_1, \dots, p_n = p_e$) and $p_i/m-1$, respectively.

$$y_{\text{uniform}} = \begin{pmatrix} 1 - p_1 & \cdots & p_1/m-1 \\ \vdots & \ddots & \dots \\ p_n/m-1 & \cdots & 1 - p_n \end{pmatrix} \tag{3}$$

Natural distribution An approach that is less considered in the literature but may occur in practice is the natural distribution case, where the noise distribution is proportional to the natural distribution of the remaining classes. That is to say that, when a noise for a certain class occurs, another class with a probability proportional to the natural class

distribution replaces it. A possible situation where a natural noise distribution occurs is when the class is derived from the readings of a sensor (e.g., the graduation of a disease based on a screening test). A random Gaussian noise in the sensor would generate a natural distribution in the class noise. The labeling matrix is shown in Eq. 4. For the NCAR and NAR cases, the noise probabilities for each class are $\gamma ij = (\pi_j/1-\pi_i)p_e$ (as $p_1, \ldots, p_n = p_e$) and $\gamma ij = (\pi_j/1-\pi_i)p_i$, respectively, where π_i, π_j are the class distributions for $i, j, i \neq j$.

$$y_{\text{natural}} = \begin{pmatrix} 1-p_1 & \cdots & (\pi_j/1-\pi_i)p_i \\ \vdots & \ddots & \cdots \\ (\pi_j/1-\pi_i)p_i & \cdots & 1-p_n \end{pmatrix} \tag{4}$$

Default class In the default class case, whenever a noise label occurs, and regardless of its true value, it is incorrectly assigned to one of the classes. This type of noise may occur when, for instance, a default value (e.g., "other") is assigned when the true class could not be determined. Let c_d be the default class. A labeling matrix, for (an arbitrarily chosen) default class 3, is shown in Eq. 5. Then, for NCAR and NAR, $\gamma ij = p_e$ and $\gamma ij = p_i$ if $j = d$, and 0 otherwise.

$$y_{\text{default}} = \begin{pmatrix} 1-p_1 & 0 & p_1 & \cdots & 0 \\ 0 & 1-p_2 & p_2 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & p_n & \cdots & 1-p_n \end{pmatrix} \tag{5}$$

Blockwise In the blockwise case, there are correlations among the noise patterns of some classes, so that they form sub-matrices (or blocks) within the transition matrix. A common pattern studied in the literature is the pairwise case [147,149], in which two classes c_1 and c_2 have their true class mislabeled with some probability (for instance, in handwritten digit classification, the digits 3 and 8 have similar shape and may be mislabeled at same rate in the data set). In this case, only two cells outside the diagonal of the labeling matrix are nonzero. However, blocks may have more than two class involved. Furthermore, some data sets may also have more than one block (for instance, in handwritten digit classification, two pairwise blocks may exist, e.g., digits 3–8 and 1–7), and the blocks may have different noise patterns (e.g., one can be NAR and the other NCAR). A labeling matrix with two (arbitrarily chosen) blocks is shown in Eq. 6.

$$y_{\text{block}} = \begin{pmatrix} 1-p_1 & p_1 & \cdots & 0 & 0 \\ p_2 & 1-p_2 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1-p_{n-1} & p_{n-1} \\ 0 & 0 & \cdots & p_n & 1-p_n \end{pmatrix} \tag{6}$$

To epitomize the necessity to take into account the peculiarities of multiclass cases, we use artificially generated data sets.[1] The data sets are created with 5 classes and two attributes, with class centers evenly distributed alongside a circle of ratio $\sqrt{\#\text{of classes}}$ ($\sqrt{5}$), and a standard deviation of 1. We have data sets with four different class distributions. The class distributions were generated according to a geometric progression which sums up to 1 and

[1] We opt to use artificially generated data to have control about data characteristics.

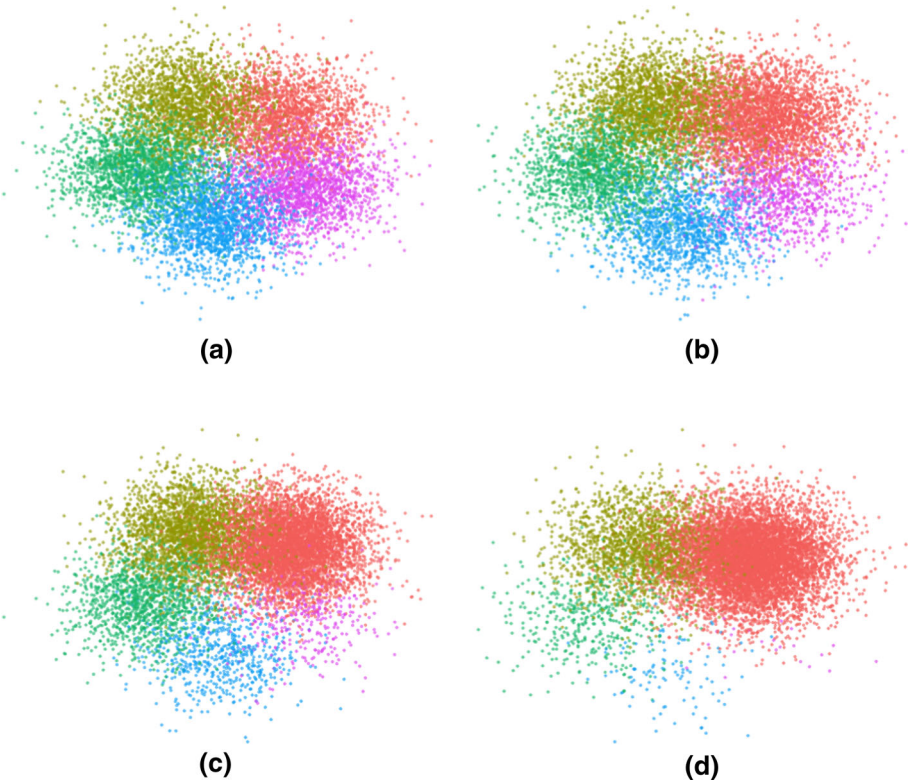

Fig. 3 Pictorial representation of the artificial data sets. **a** Ratio = 1. **b** Ratio = 0.75. **c** Ratio = 0.5. **d** Ratio = 0.25

four different ratios: 1, 0.75, 0.5 and 0.25. These ratios were selected to generate data with different class distributions, as shown in Fig. 3. A ratio of 1 means that class proportions are equally distributed for all classes (so we have 2,000 points for each class), whereas a ratio of 0.25 means that the second most frequent class has 1/4 of the instances of the most frequent class, the third most frequent class has 1/4 of the instances of the second most frequent and so on (so the number of instances in classes are about 7500, 1875, 472, 121, 30). A pictorial representation of the four data sets is shown in Fig. 3, where colors indicate class values. Similarly, for ratios of 0.75 and 0.5, the number of instances reduces in a proportion of 3/4 and 1/2, respectively.

Figure 4 illustrates the difference in behavior, in terms of balanced error rate, of four different learning algorithms (J48 decision tree, k-nearest neighbors with k=5, linear discriminant analysis and naïve Bayes), where we artificially introduce class noise in ratios of 2.5% to 50% of instances, with steps of 5%, with the uniform and natural distribution class noise patterns. These experiments were performed using tenfold cross-validation, and artificial noise is introduced only in the training set. As expected, there are no perceptible differences when classes are equally balanced (the "1" column). In this case, both corrupting the class uniformly or taking into account the natural class distribution has virtually the same effect, and the natural class distribution is uniformly distributed. However, as the class proportion becomes more skewed, the artificial noise introduction following the natural class

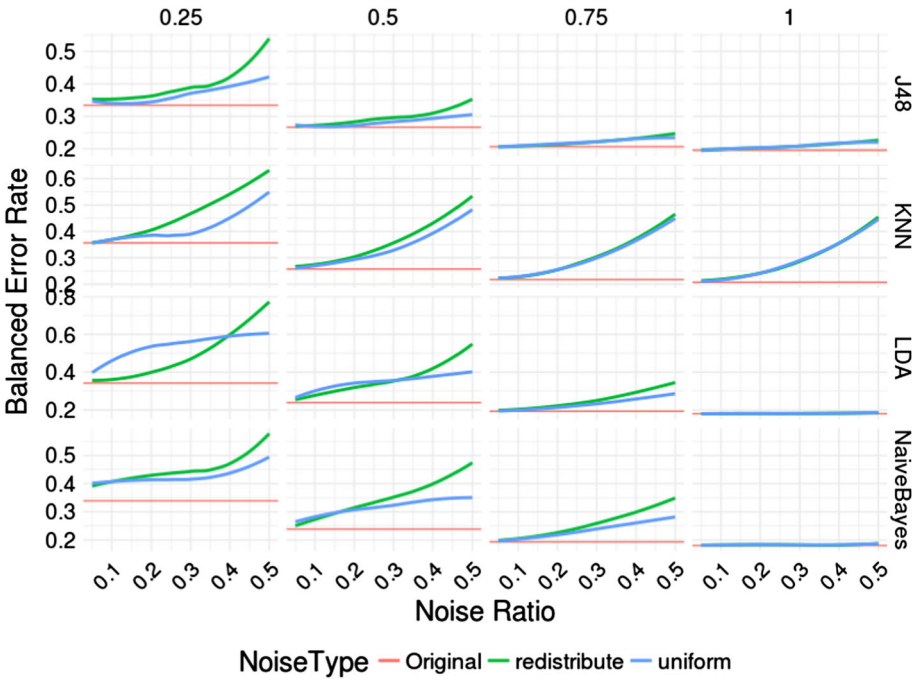

Fig. 4 Difference between uniform and natural multiclass noise pattern in an artificial data set

distribution tends to be more harmful to learning algorithms than noise introduced uniformly (except for kNN), for the same noise level.[2]

Although the influence of different noise patterns in multiclass classification is clear, these aspects are seldom discussed in the literature. To illustrate this problem, we ran INFFC [113] and CNC-NOS [72], two recently proposed techniques for treating noise, using the same experimental setup described earlier. These results are depicted in Fig. 5, where uniform noise introduction is shown in Fig. 5a, and introduction of noise according to natural class proportion is shown in Fig. 5b. Even though we are dealing with a simple problem, it is possible to see a difference in the performance of the two treatment techniques considering the two different scenarios. INFFC and CNC-NOS can reasonably recover from the noise when it is introduced following a uniform introduction pattern in almost all cases. (The lines corresponding to treated data are almost flat in most of the cases, indicating that the methods can successfully recover from increasing noise ratios.) However, we can observe that the treatment methods are less effective when noise is introduced accordingly to a natural class distribution for highly imbalanced data sets, where the differences in class distribution are larger. (The lines corresponding to treated data bend upwards for larger noise rates, for the data sets generated with 0.25 and 0.5 ratios from minority to majority classes.) This experiment shows that even state-of-the-art methods present different behavior for different types of noise patterns.

The multiclass case can also imply further issues in the NNAR case (in which the class noise also depends on the input space). Some class combinations may be differently influ-

[2] In this particular simple problem, some algorithms are more resilient to class noise for low noise rates, although we can observe this trend for higher noise rates.

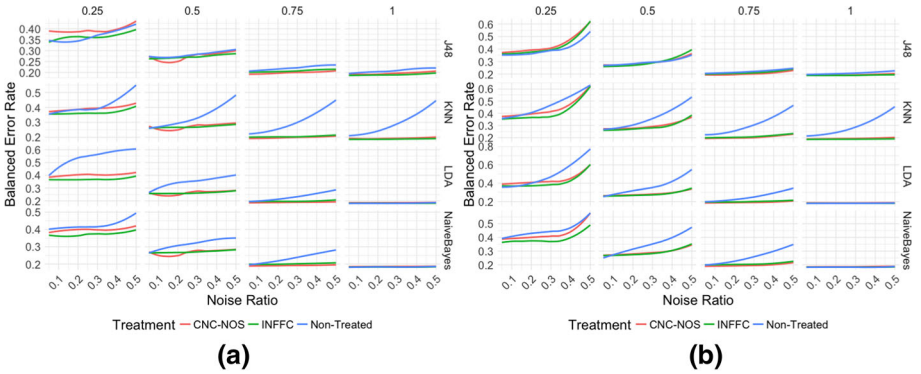

Fig. 5 Comparison of treated (using INFFC and CNC-MOS) and untreated data. **a** Uniform noise introduction. **b** Natural noise introduction

enced by different regions in the input space. There may be regions in the input space, for example, where the likelihood of class noise for some class combinations is higher than others, while in other regions other class combinations are more likely to be confused. For example, different class boundaries may influence the likelihood of class noise in different regions in the feature space, implying in different blockwise interactions in different regions of the space. Another example that may occur is in high density regions, which may be more likely mislabeled as majority classes or a default class, for instance.

Figure 6 shows a similar experiment to Fig. 4. However, the likelihood of an instance to be mislabeled increases with the distance to the class center. Although the shapes in both figures are similar, we can observe an increase in the error rate of the algorithm, especially J48.[3] It is expected that in more challenging multiclass classification tasks also becomes much more challenging.

We repeat the experimental setup of Fig. 5 using the NNAR noise setup to introduce artificial noise for the experiments shown in Fig. 6. It is interesting to observe that the noise treatment techniques behave differently in this case for the natural and uniform class distribution patterns. CNC-MOS is much less effective for alleviating the noise problem. When the class distribution is more skewed (ratios 0.5 and 0.25), CNC-MOS sometimes is worse than non-treating the data. Furthermore, for natural noise patterns, the performance degradation occurs at lower noise rates for CNC-MOS. The degradation in performance can be explained by the fact that CNC-MOS was designed with the assumption of NCAR noise, but this difference between uniform and natural noise could be studied further.

3.2 Noise multilabel classification

In multilabel classification, an instance can be assigned, at the same time, to multiple target labels.[4] For instance, given a set of topics in a news outlet, a news article can be classified as both sport and politics if it is concerned about rioting against the exuberant cost of funding the World Cup. More formally, given a set $\mathcal{Y} = \{l_1, \ldots, l_m\}$, instances are associated with a subset $Y \subseteq \mathcal{Y}$, where the cardinality of Y can be any value between 1 (i.e., $Y \neq \emptyset$) and

[3] A possible explanation for this behavior of J48 is related to the uncertainty in leaf boundaries in the tree

[4] Multilabel classification should not be confused with multiclass classification (Sect. 3.1), which is the problem of categorizing instances into one of more than two classes.

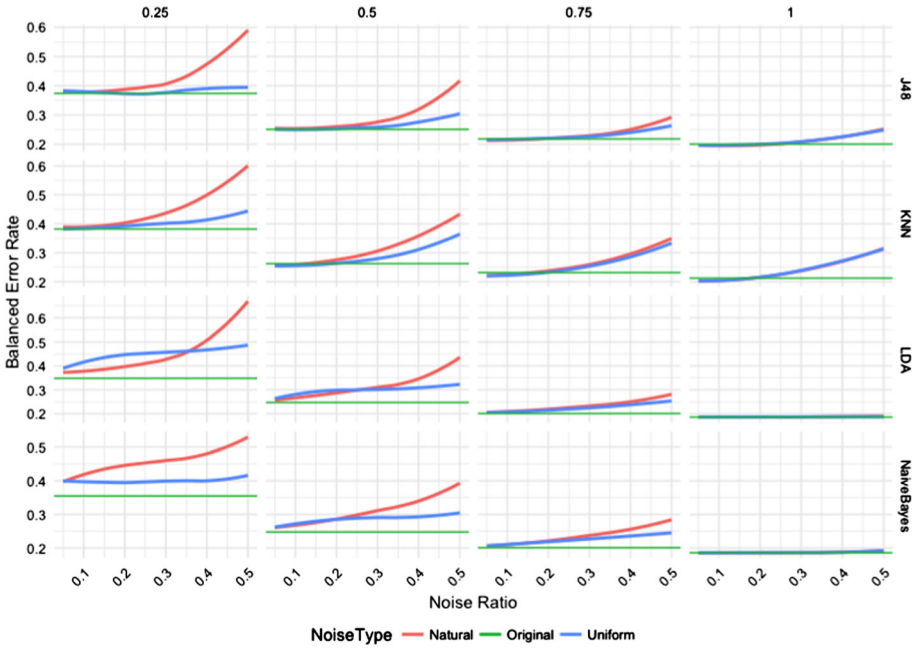

Fig. 6 Difference between uniform and natural multiclass noise patterns in an artificial data set, where instances far from the class center are more likely to be mislabeled

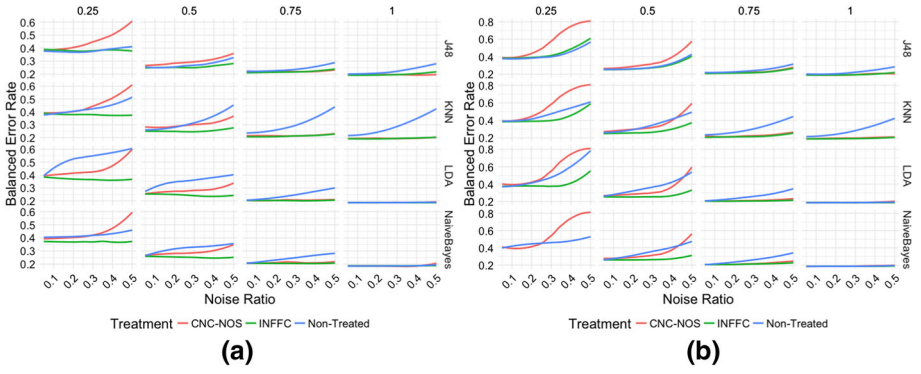

Fig. 7 Comparison of treated (using INFFC) and untreated data. **a** Uniform noise introduction. **b** Natural noise introduction

the number of labels n_l. The set Y is called the set of relevant labels, while the complement set $\mathcal{Y} \backslash Y$ is considered irrelevant for that instance. To facilitate notation, we define a binary vector $\mathbf{y} = (y_1, \ldots, y_m)$, where $y_i = 1$ indicates the presence (relevance) and $y_i = 0$ the absence (irrelevance) of l_i in the labeling of a instance (Fig. 7).

Dealing with noise in multilabel classification is a very important topic, as numerous applications use "soft labels": labels which are not assigned by a domain expert but are derived from automatic taggers [101] or from non-experts via crowdsourcing [92], which are known to introduce noise. However, the influence of noise in multilabel classification has so far been little studied. For each instance, multilabel noise may occur in two situations: (a)

an incorrect false label is included in the relevant label set or (b) a correct true label is not included in the relevant label set.

A straightforward approach for dealing with multilabel noise is to apply data transformation methods [45,51] and deal with noise in the transformed data sets. Two possible approaches are considering each label individually (i.e., using a strategy similar to binary relevance and consider a binary noise problem for each label) or each label set (i.e., using an strategy similar to label power sets and consider a multiclass problem for the possible label sets) for dealing with labels. This approach will transform the problem of dealing with multilabel noise to binary or multiclass noise problems, so that methods which have been natively proposed to these problems can be applied to handle multilabel noise.

Dealing with the absence of relevant labels is also challenging. Some labels may be very rare (i.e., they suffer from a "label imbalance" problem, a class imbalance problem in multilabel context [16]). Filter techniques were not directly applicable in this case, and noise reparation methods may have poor performance due to the level of scarcity of some labels, as those techniques are generally data driven and cannot work well under the absence of information.

Another problem that may have influence on noise patterns in the multilabel case is related to label dependence. In several multilabel applications, labels are correlated, i.e., the occurrence/absence of one label may influence or be influenced by the occurrence/absence of other(s). For instance, multilabels can be organized into a hierarchical [127] or a graphical [123] structure. Furthermore, this structure may or may not be known at training time [87].

There are two types of independence to consider [24]: marginal independence and conditional independence. When the marginal independence is violated, there are global correlations among labels that are not affected by the input space. In other words, the product of the marginal distribution of labels could not characterize the overall distribution of labels [24]. On the other hand, a conditional independence violation occurs when the correlation among labels is concentrated in some regions of the input space. This is to say that some data characteristics increase or decrease the co-occurrence of labels.

Therefore, when considering label patterns (e.g., for experimental simulations on multilabel classification), there are three cases to be considered:

Individual Label Noise In which noise patterns are considered in isolation for each label, and the inclusion of noise in one label does not influence the occurrence of noise in other labels. As each label is considered in isolation, individual label noise for each label pattern could be assumed to be NCAR, NAR or NNAR. In the NCAR case, for a particular label, the noise probability for relevance and irrelevance is the same. In NAR, different noise probabilities for relevance and irrelevance could be modeled, and for NNAR, the relevance and irrelevance noise probabilities also depend on the input space. Furthermore, as labels are binary (it is either relevant or not), a noise occurrence for one label implies in switching the value of that label for an instance from 0 to 1, or vice-versa.
Joint Label Noise In which noise patterns take into account correlations among labels. The stronger the correlation, the higher the probability of different correlated labels for an instance to be corrupted together. This is likely to occur, for instance, if the labeling process follows a hierarchical process. If an error occurs in a higher level in the hierarchy, it will propagate to lower levels. To model this dependence, a label correlation model should be taken into account. A model widely adopted in the literature for balancing model complexity and capacity is a sequential chaining structure [19,87]: l_i is conditioned on $\{l_1, \ldots, l_{i-1}\}$. This model is based on the product rule of probability and is the basis of classifier chains [108] for dealing with model dependence. A possible way to model error

in the label chain is shown in Eq. 7.

$$p\left(\tilde{l}_i, e|\{l_{i-i}, \ldots, l_i\}\right) = p\left(\tilde{l}_i|\{\tilde{l}_{i-i}, \ldots, \tilde{l}_1\}\right) \times p\left(e|\{\tilde{l}_{i-1}, \ldots, \tilde{l}_1\}, \tilde{l}_i\right) \quad (7)$$

The first term is due to the label chaining and refers to the influence of previous labels in the definition of the actual label. The second term refers to the probability that an error in the labeling occurs, conditioned to the label chain. Given the label chain, this probability may not depend on the actual label or the instance space, a case analogous to NCAR; it depends only on the relevance/irrelevance of the actual label, a case analogous to NAR, or depends both on the current label and instance space, an analogous situation to NNAR.

Joint Instance and Label Noise In which noise patterns take into account correlations among the label and the input space. This situation is similar to the Joint Label Noise, except for the fact that a label correlation should also depend on the instance space. A possible formulation considering the chain rule is shown in Eq. 8.

$$p\left(\tilde{l}_i, e|\{\mathbf{x}, l_{i-i}, \ldots, l_i\}\right) = p\left(\tilde{l}_i|\{\mathbf{x}, \tilde{l}_{i-i}, \ldots, \tilde{l}_1\}\right) \times p\left(e|\{\mathbf{x}, \tilde{l}_{i-1}, \ldots, \tilde{l}_1\}, \tilde{l}_i\right) \quad (8)$$

The second term models the error, considering the correlation between labels and the instance space. Given a label chain that also depends on the instance space, we may have cases analogous to NCAR, NAR and NNAR.

A problem with considering a chaining model for the label correlation is that an ordering among labels must be chosen. This problem has been tackled with a combination of classifiers using different orderings [19]. It would be interesting to analyze the interactions of these methods with noise. Another problem is that, for problems with a large number of labels, the correlation model could be quite complex. A possible approach to overcome this is to consider only correlations within subsets of labels, such as in [134], as it is unlikely that all labels influence each other.

Figures 8 and 9 show the behavior of multilabel classifiers in artificially generated data sets. The artificial data set generation is as follows: 10,000 data points are generated into five clusters (the size of the clusters is similar), with two attributes generated similarly to Fig. 3. However, clusters are labeled from a set of five possible labels. The probability of each label is generated according to a geometric progression, with ratios of 1, 0.75 and 0.5, meaning that labels are equally probable (ratio equal to 1) to the probability of the most frequent label being twice as higher as the probability of the second label, and so on. The label cardinality is about two, meaning that on average two out of five labels are present. For each cluster, the probability of one of the labels is set to zero (meaning this label is not present in that cluster).

Figure 8 shows that the label correlation is zero (that is, the occurrence of one label does not influence the occurrence of others), while the label correlation in Fig. 9 is 0.5, according to the procedure described in Leisch et al. [63]. Noise was introduced independently (individual label noise) and taking into account label correlations (positive labels have 90% chance of being corrupted together when there are more than two positive labels in the same instance). Four different multilabel learning algorithms were utilized. Three of them use a problem transformation approach, converting the multilabel problem to a set of single-label problems and using standard binary classifiers (we use J48 and kNN as base classifiers). Binary relevance (BR) converts the multilabel problem to independent binary classification problems. Classifier chain (CC) trains a chain of classifiers where in each step the data are augmented with previously trained labels. Dependent binary relevance (DBR) uses the values of all other labels. Finally, rFerns Ozuysal et al. [94] uses a version of random forests adapted to the multilabel case. Experiments were carried out using tenfold cross-validation.

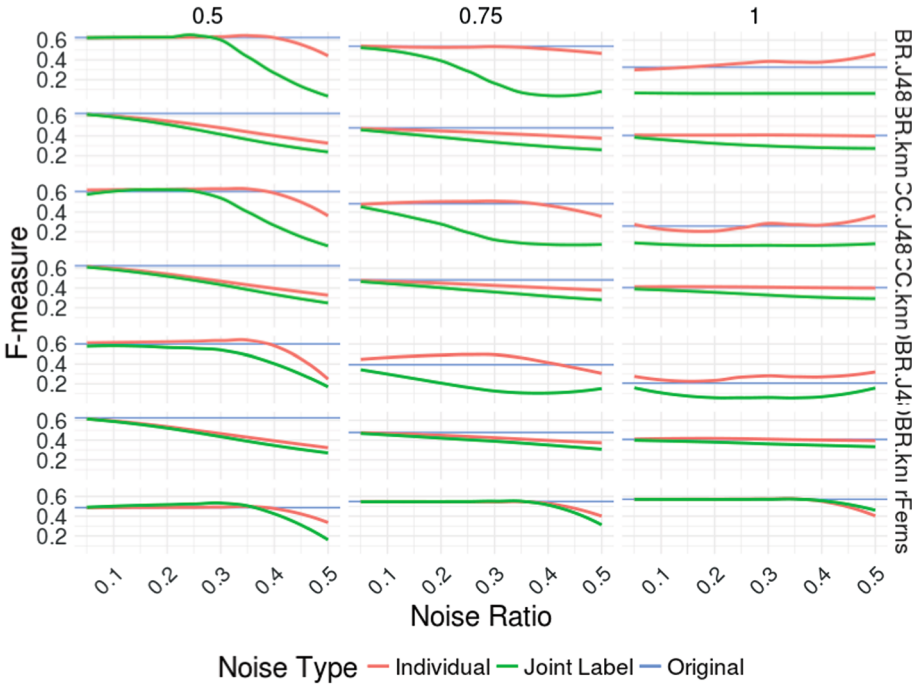

Fig. 8 F1-measure for a multilabel data set with uncorrelated labels

Analyzing the figures, it is interesting to note that when labels are correlated, the noise introduction pattern which jointly considers positive labels has a stronger negative effect on the behavior of learning algorithms (measured in terms of the average F1-measure over all labels) when labels are correlated. This phenomenon is even accentuated for algorithms designed to take label correlation into account (CC and DBR). The results in this simple artificial data set illustrate the importance of considering joint label noise patterns in research with noise in multilabel problems.

3.3 Noise in multitask problems

Multitask learning [13] is a machine learning paradigm which aims to improve the performance of learning algorithms by learning classifiers for multiple tasks together, potentially exploiting commonalities between them. More formally, given a set of supervised tasks $\mathcal{T} = \{T_1, \ldots, T_m\}$, where for each task T_i a set of examples $\{X_i, Y_i\}$ is available. The goal is to learn a set of m models simultaneously $\mathcal{H}_i : X_i \rightarrow Y_i$ from the available examples.

The main argument in favor of multitask learning is that the relations between the tasks make it advantageous to learn the tasks simultaneously instead of learning each task independently. For instance, one can learn web-ranking functions for different languages were each language represents a task. Although it is possible to learn an independent ranking function in isolation for each language, by learning them simultaneously it is possible to explore cross-lingual information that enhances the rankings [15]. Another example is the prediction of disease progression measured by the cognitive scores, by learning a predictor simultaneously at multiple time points [145].

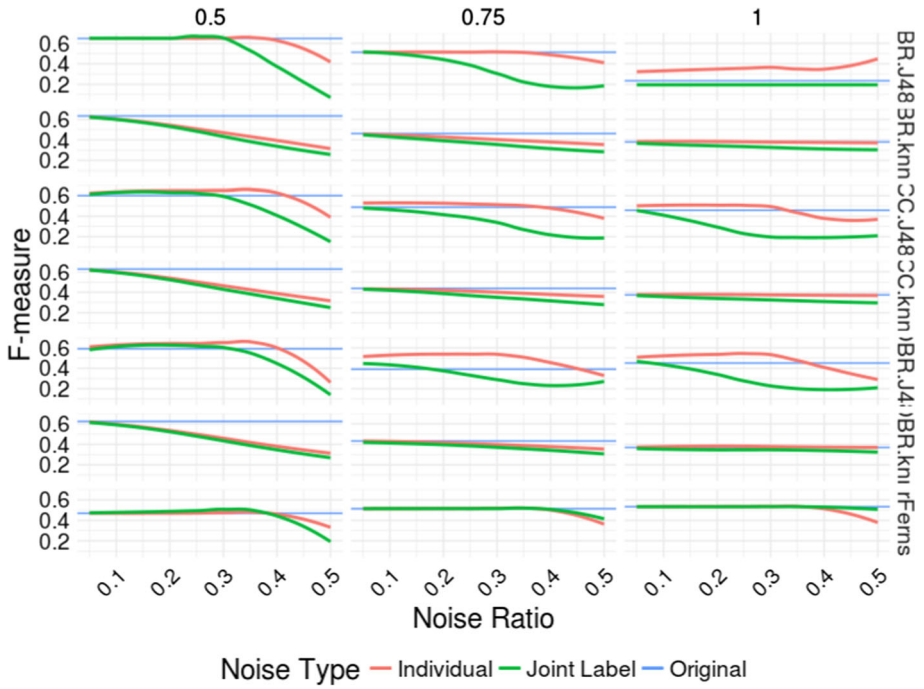

Fig. 9 F1-measure for a multilabel data set with correlated labels

There are some options to explore the commonalities. The different tasks may share some features (for instance, data that come from a relational database may have some features in common), or similar feature extraction techniques can be used (the same algorithm could be used to extract features from images [139], for instance). Another possibility is that the same object is described by different set of uncorrelated features (multiview learning [121]), and the different views are used to improve classification [10].

A straightforward approach for (artificial) noise modeling is to consider an independent noise modeling for each task. In this scenario, different tasks may have different noise patterns, which can be NCAR, NAR or NNAR. However, noise simulation patterns could also take into account the commonalities among the tasks under consideration. For instance, when tasks share the class information (as in multiview learning cases), the same noise pattern should be considered for all tasks.

An area of research closely related to multitask learning is domain adaptation or transfer learning [95], where a source domain, generally with a large training data, is used to help the learning of a target domain, where very little training data is available. For instance, it is possible to use data from a city where a large labeled data set is available to improve predictions in other cities [129]. It would be interesting to investigate transfer learning approaches for noise handling.

Figure 10 shows a comparison of different noise scenarios with and without transfer learning. For this experiment, data from source and target domains were generated according to the settings 1 and 2 from Friedman [31], respectively, using the `klaR` package.[5] Source domain contains 5000 data points, while target domain 2000 data points. The number of

[5] https://cran.r-project.org/package=klaR.

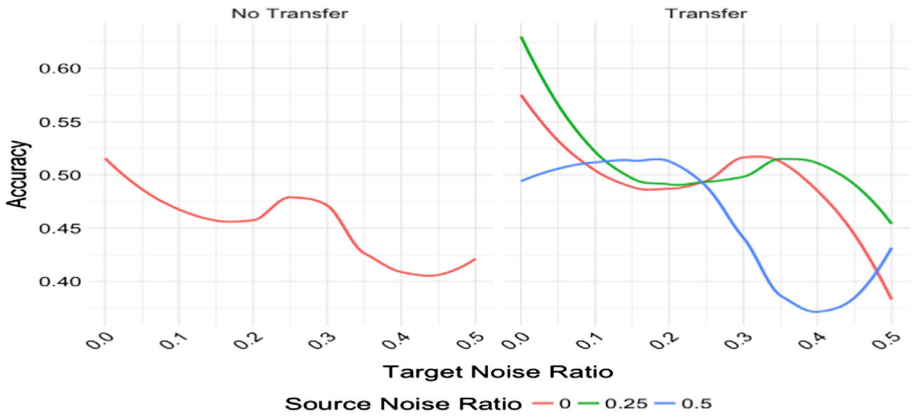

Fig. 10 A comparison of different noise scenarios with and without knowledge transfer

classes in both domains is three, and the number of features is six. Source domain uses equal spherical covariance matrices, whereas target domain uses unequal spherical covariance matrices. In the figure, the x-axis corresponds to uniform artificial noise introduction in the target, ranging from 0 to 50% with steps of 5%. We use the same procedure for artificial noise introduction in the source domain, although for the sake of visualization, only the cases of 0, 25 and 50% are shown in the figure. We use the incremental version of stochastic gradient descent approach for learning support vector machines from the `sklearn` package. The figure shows accuracy, estimated using a fivefold cross-validation, with and without knowledge transfer from tasks. The model is initialized with source data and then tuned with target data. Analyzing the graph, we can observe that the benefit of domain adaptation is reduced when the source is corrupted with noise. Furthermore, the combination of large noise rates in both source and target domains is worse than using only the target domain.

3.4 Noise in multi-instance learning

In the multi-instance learning framework, training examples consist of "bags" (collections of instances) rather than singleton instances [2]. This allows the representation of complex problems by means of decomposing them into pieces [20]. For example, consider the case of image classification. Multiple patches can be extracted from an image, whereas each patch is associated with an instance. Thus, a bag containing all the patch instances can be used to represent an example (image). Another example is text classification. Sections of a document can be represented by instances, whereas the document itself is represented as the bag of these instances.

The typical formulation assumes that bags belong to two classes (generally called positive and negative). Furthermore, only the labels of the bags are known (i.e., the individual labels of the instances within each bag are unknown). For example, in object recognition labels are associated with an image. The presence of an object is known, while its exact location within the image is not. A bag is labeled as positive if it contains at least one positive instance and negative otherwise. More formally, pairs $\{\mathcal{B}_i, y_i\}$ compose a data set, where $\mathcal{B}_i$ is a bag, and y_i is its class. Moreover, each $\mathcal{B}_i$ is composed by a set containing j instances $\{\mathbf{B}_{i1}, \ldots, \mathbf{B}_{ij}\}$ (the index i represents a bag, and the index j instances within the bag) and, generally, the

class $y_i \in \{0, 1\}$. Although multiclass labels may be used [96,136], the use of multi-instance multilabel learning is also common [146].

In multi-instance classification, class noise may occur when a bag is incorrectly labeled. In the binary case, this means that a bag that does not have any positive instance is incorrectly labeled as positive, or when a bag which has at least one positive instances is incorrectly labeled as negative. However, it is also possible to analyze noise at the instance level. Multi-instance learning 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 [67]. A common approach is to estimate the diversity density estimator [76], where the objective is to build a model which takes into account the evidence required to classify positive instances from the union of positive bags, filtering out its intersection with negative bags.

The noise-OR approach [97] assumes that there is at least one positive instance inside a positive bag and no positive instances in the negative bags. This approach considers instances within a bag as independent and computes the probability of a class given the positive bags using the noise-OR, as shown in Eq. 9:

$$p\left(c_i|\mathcal{B}^+\right) \propto 1 - \prod\left(1 - p(\mathbf{B}_{\mathbf{ij}}^+ \in c_i)\right) \tag{9}$$

where $p(\mathbf{B}_{\mathbf{ij}}^+ \in c_i)$ is the probability of instances in a positive bag $\mathcal{B}_i^+$ of being positive and can be calculated in several ways. In [64], a boosting approach was used to compute $p(\mathbf{B}_{\mathbf{ij}}^+ \in c_i)$ for video taxonomy classification, in order to cope with noise positive bags. Possible noise negative bags were ignored, as the authors argue that this type of noise can be negligible in this kind of application.

A similar approach to noise-OR but with few independent assumptions is to consider the most likely instances within each bag [76]. In Li and Vasconcelos [66], the idea of considering the most likely instances (which authors call top instances) was used to develop a large margin classifier for multi-instance classification. Their approach is applicable to positive and negative bags and thus handles both positive and negative noise bags.

The modeling of different noise scenarios should consider how bags are formed. If bags consist of the creation of instances from labeled examples (e.g., bags are composed of chunks from pre-labeled images), the noise scenarios could ignore the influence of instances within a bag and consider only the noise at bag level. From this perspective, similar approaches to binary, multiclass or multilabel cases can be considered, depending on the type of bag labels.

Figure 11 shows results of an artificially generated data set using `milr`[6] R package Chen et al. [18]. The data were generated with 2000 bags. Each instance is generated with 10 features randomly generated with a mean of zero and a standard deviation 1. The number of instances within each bag was sampled uniformly from the range 2–5. The label of each instance was determined according to a logit link, with the parameter β randomly drawn from -5 to $+5$, independently for each feature. A bag is labeled as positive if it contains at least one positive instance. Noise was introduced in two ways: at bag level and at instance level. At bag level, bag labels are flipped with probability p, while at instance level, instance labels are corrupted at with probability p. Changing the label of instances may change the label of a bag if all positive instances of a bag positive bag are corrupted to negative, or if a single negative instance is corrupted to positive.

The bag classification error (estimated using fivefold cross-validation) of five different multi-instance learning algorithms is compared in Fig. 11: CitationKNN (uses the Hausdorff

[6] https://cran.r-project.org/web/packages/milr/.

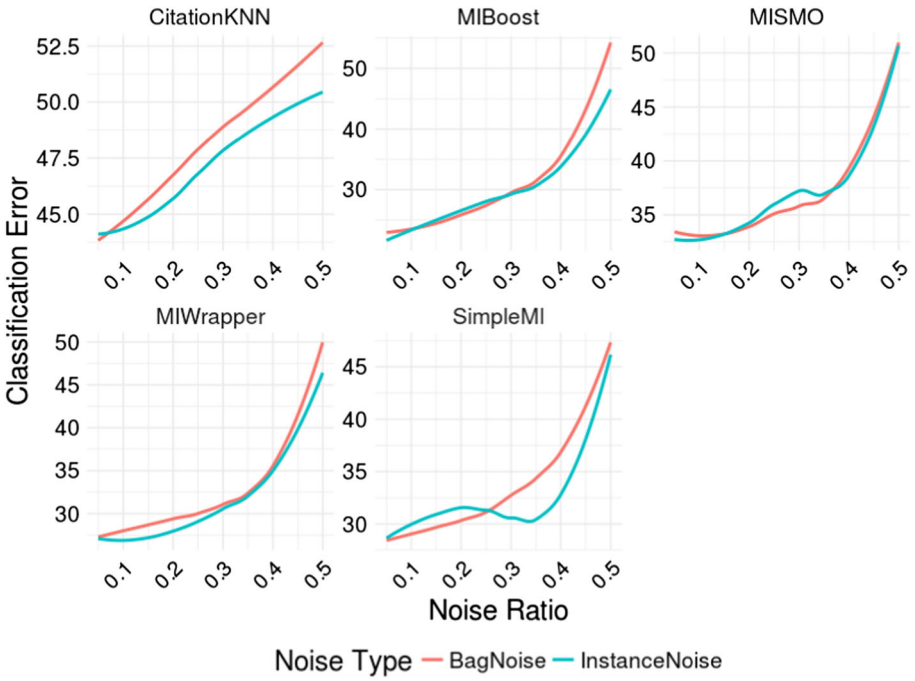

Fig. 11 A comparison of bag noise and instance noise in multi-instance classification

distance and the concept citation to find the bag nearest neighbors), MiBoost (boosting adaptation to multi-instance), MiSMO (sequential minimal optimization for SVM adaptation for multi-instance, MiWrapper (combines the scoring prediction for each individual instance within a bag) and SimpleMI (converts the multi-instance problem to a single instance problem by computing the average of all instances within a bag). Although for some algorithms the two different noise types show a similar degradation performance, in CitationKNN and SimpleMI there are some notable differences. The degradation of instance noise is lower than bag noise for CitaitonKNN for almost all artificially generated noise levels, while for SimpleMI, although it is a little higher at lower noise levels, it is lower for intermediate noise levels.

However, in some multi-instances problems, bags are not formed from a single object, but from joining instances that have some characteristics in common. For instance, in [77] multi-instance learning was used for stock market prediction. Positive bags where formed from stocks which had an increase in value from an instant $t-1$ to the instant t and negative bags from stocks that did not show an increase. Bags are labeled positive by the assumption that the increase in *some* stocks are due to some fundamental reasons and would lead to an increase in value at some instant $t+1$, whereas negative labeled bags assume that a decrease in value of *all* stocks are due to some fundamental reasons and would not lead to an increase in value at a future instant $t+1$. Both assumptions are unlikely to hold for all time stamps; thus, noisy labeled bags can occur. Furthermore, these noisy bags may depend on the instances that form the bags. For instance, the stock market may be in high volatility period, and (most of) the stocks are going up or down during this period, irrespective of their fundamental

reasons. It would be interesting to analyze approaches for dealing with bag noise that takes into account noise due to bag formation.

3.5 Noise in ordinal classes

For some machine learning tasks, classes have a natural ordering relation [48]. For instance, a user can use the ordinal scale {*bad*,*average*,*good*,*very good* and *excellent*} for a movie review. More formally, the $\mathcal{Y} = \{c_1, \ldots, c_m\}$ has an ordering constraint $c_i \prec c_j, \forall i < j$, where $\prec$ is an order relation. A common assumption is that data are arranged according to an ordinal scale, i.e., only the ordering among classes matters, not their relative differences. In other words, the classes do not carry metric information, and they only serve as a qualitative indicator.

A naïve approach to define a class noise in ordinal problems is similar to that for multiclass problems (Sect. 3.1), where a class c_i is erroneously replaced by another class in $\mathcal{Y}\backslash\{c_i\}$. However, this definition considers each mislabeling as equally damaging. Nevertheless, due to inherit ordering among classes, some mislabels are more harmful than others, as mislabeling of adjacent labels may have a lower impact on the learning phase than the mislabeling of distant labels.

In this sense, it would be interesting to investigate cost-sensitive noise handling techniques. Indeed, cost-sensitive learning is one approach for ordinal classification [125] and some approaches such as cost-weighted binary decomposition [68] could be used to extend binary class noise dealing techniques. Another approach for extending binary noise handling techniques to ordered case is to exploit ordinal decomposition [48]. Instead of *one-versus-all* and *one-versus-one* decompositions, common in multiclass classification, ordered decomposition uses i binary subproblems by: *ordered-partition* ($pos = \{c_j | j > i\}$ and $neg = \{c_k | k \leq i\}$), *one-versus-next* ($pos = \{c_i\}$ and $neg = \{c_{i+1}\}$), *one-versus-follows* ($pos = \{c_i\}$ and $neg = \{c_j | j > i\}$) and *one-versus-previous* $pos = \{c_i\}$ and $neg = \{c_j | j < i\}$ decomposition.

Ordinal classes may require the ordering to be included in noise modeling. Although, as in multiclass cases, arbitrary patterns may be used, some of them being of practical interest. Let $\mathcal{P} = \{p_1, \ldots, p_m\}$ be the mislabeling probability of each class,[7] and γ_{ij} the probability of a instance of class i be mislabeled as class j for $i, j = \{1 \ldots m\}$, and let $d_{ij} = |i - j|$ the distance (in ordinal scale) between classes c_i and c_j. Some noise patterns of interest include:

Adjacent label noise In this pattern, adjacent classes are more likely to have their labels mixed. A possible adjacent noise distribution is to consider a linear decay approach, as can be defined as $\gamma_{ij} = {}^{m-d_{ij}}/_{S}$, where $S = \sum_{j=1}^{m} d_{ij}$. Another possibility is to consider a square decay approach where the probability of a instance of class i being mislabeled as class j is $\gamma_{ij} = {}^{(m-d_{ij})^2}/_{S^2}$ where $S^2 = \sum_{j=1}^{m} d_{ij}^2$.

Diametrical label noise In this pattern, diametrical (opposite) classes are more likely to have their labels mixed. Linear and square decay approach could be defined as $\gamma_{ij} = {}^{d_{ij}}/_{S}$, and $\gamma_{ij} = {}^{d_{ij}^2}/_{S^2}$, respectively.

Optimistic label noise In the optimistic case, the probabilities of a class i being mislabeled as class j is higher for $j > i$ in comparison with $j < i$. One possibility is to adopt fixed rates $p_{\succ} > p_{\prec}$ where $\gamma_{ij} = p_{\succ}$, $j > i$ and $\gamma_{ij} = p_{\prec}$, $j < i$. Other possible approaches are the adaption of adjacent and diametrical label noise, restricted to cases where $j > i$ and setting $\gamma_{ij} = 0$ for $j < i$.

[7] This corresponds to noise at random. For the noise completely at random case, we have $p_1 = p_2 = \ldots = p_n = p_e$.

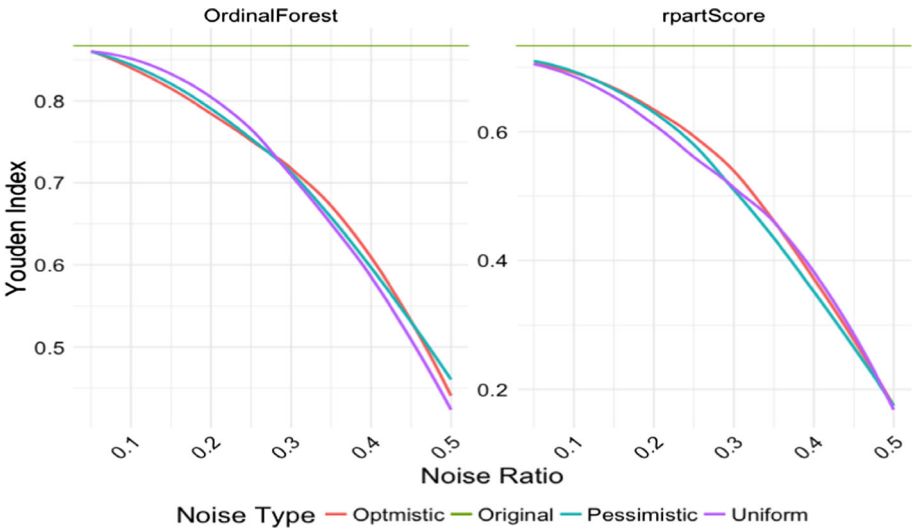

Fig. 12 A comparison of optimistic, uniform and pessimistic noise in ordinal classification

Pessimistic label noise Pessimistic cases are just the opposite to the optimistic cases, i.e., the probabilities of a class i being mislabeled as class j is lower for $j > i$ in comparison with $j < i$. One possibility is to adopt fixed rates $p_{\succ} < p_{\prec}$ where $\gamma_{ij} = p_{\succ}$, $j > i$ and $\gamma_{ij} = p_{\prec}$, $j < i$. Other possible approaches are the adaption of adjacent and diametrical label noise, restricted to cases where $j < i$ and setting $\gamma_{ij} = 0$ for $j > i$.

A special case within ordinal classification is monotonic classification, where there are monotonic constraints between (some) features and an ordinal class [47]. For instance, the credit range of a bank customer may be monotonic depending on its income and patrimony. It has been argued that properly taking into account such constraints may improve classification performance in monotonic classification problems [7]. Instances violating the monotonic constraints can also be considered noisy [22,54]. The generation of noise monotonic data sets is data dependent, and some approaches can be found in Daniels and Velikova [22,84].

Figure 12 shows a comparison between two ordinal classifiers for an artificial data set. Data were generated according to a bivariate Gaussian distribution, with mean 0 and standard deviation 1, with 1000 data points. Data are divided into five classes according the radial distance to data centroid. The first ordinal class is the closest to the centroid, whereas the last ordinal class corresponds to that furtherest from the centroid. We simulate three types of noise: uniform, where the corrupted class instance can assume the values of other classes with equal probability; optimistic, where a noisy instance has a 90% probability of being corrupted to a greater class value (when possible) and pessimistic, where a noise instance has a 90% of probability of being corrupted to a lower class value (when possible). Two ordinal classifiers were compared: `rpartScore` [35], an ordinal classifier based on decision trees, and `ordinalForest` [53], an ordinal classifier based on random forests. The figure shows the Youden agreement index, a metric commonly used in ordinal classification. Analyzing the figures, it is interesting to observe that the performance of the uniform artificial noise inclusion is somewhat between optimistic and pessimistic artificial noise inclusion. Furthermore, the performance with pessimistic and optimistic noise "crosses" when the percentage of artificial noise increases. This reflects a possible trend of the ordinal classification algorithms in

preferring lower or greater class value predictions. Analyzing the behavior of ordinal and monotonic classification problems with different noise patterns could be an interesting topic for further research and may help shed some light on understating the properties of these algorithms.

3.6 Noisy data streams and non-stationary environments

In a data stream environment, data are provided as an ordered sequence of examples where, in general, the objective is to predict the class or value of new instances in the data stream given some knowledge about the class membership or values of previous instances in the data stream [33]. In other words, we have a sequence of instances $(\mathbf{x}_t, y_t)$, where $\mathbf{x}_t$ represents the feature values of the instance at time t, and y_t is its (unknown) class value. Data may arrive a single instance at time (online) or in sets (batch). In the online case, only a single instance $\mathcal{S}_t = \{(\mathbf{x}_t, y_t)\}$ is provided at each time stamp, whereas in the batch case a set $\mathcal{S}_t = \{(\mathbf{x}_t^1, y_t^1), \ldots, (\mathbf{x}_t^n, y_t^n)\}$ is available at each time stamp.

This streaming environment introduces several requirements for learning and has consequences for dealing with class noise. First, time and memory constraints require fast and memoryless techniques and may limit the use of some noise dealing methods. Instances may arrive at a fast pace and in general should be processed only once [42]. Therefore, noise dealing techniques that work as preprocessors should obey these constraints.

Another important aspect is dealing with the non-stationary aspect of the data [26]. Evolving data streams require adaptive methods responsible for changes in the data distribution. These changes are known as drifts and may be abrupt, gradual, incremental or reoccurring [36]. In an abrupt drift, data distribution suddenly changes (e.g., the behavior pattern of users when the release of a new web portal). Gradual drift occurs when data described by different distributions appear together in a transition period (e.g., a new product is released, but an older version continues in the market for some time). An incremental drift occurs when there are several intermediate steps between the initial and the final distributions (e.g., the behavior of citizens as the gradual implementation of new subway system is taking place). Finally, in a reoccurring drift, a concept that occurred previously may reappear (e.g., seasonal shopping patterns). Drifts may also be random or may have some predictive properties [85].

Furthermore, the changes in data distribution can also be characterized by how the posterior probability of classes $p(y_t|\mathbf{x}_t)$ (the decision boundaries) is affected. This distribution can be decomposed as:

$$p(y_t|\mathbf{x}_t) = \frac{p(y_t)p(\mathbf{x}_t|y_t)}{p(\mathbf{x}_t)}.$$

Changes in the data distribution may occur due to changes in any of the terms [40]:

- the prior probabilities of classes may vary (i.e., $p(y_{t_i}) \neq p(y_{t_j})$);
- the probability of an instance with $\mathbf{x}$ may change over time (i.e., $p(\mathbf{x}_{t_i}) \neq p(\mathbf{x}_{t_j})$);
- the class conditional probability may change (i.e., $p(y_{t_i}|\mathbf{x}_{t_i}) \neq p(y_{t_j}|\mathbf{x}_{t_j})$);

where t_i and t_j are two different time stamps. Drifts can be categorized whether they influence or not the decision boundaries: in a *virtual concept drift*, data distribution changes without affecting the decision boundaries, whereas a *real concept drift* the changes in data distribution alters class boundaries.

Distinguishing drifts from noise is an important issue in data streams [36]. It is desirable that algorithms detect and react to drifts, but be robust to noise. There are three possible situations involving noise and data streams in the non-stationary case:

Stationary noise and evolving data stream In this case, the noise generation process does not change over time, only the data distribution does. For instance, in a monitoring and control application, the monitored process may present some drift, but data come from the same (noise) sensors. Although the process that generates data is stationary, the "observed" noisy instances may vary over time if the noise is not completely random (i.e., if it depends on the class or input space). With this in mind, it would be interesting to investigate low (computational) cost approaches for reacting to drift detection and noise identification. For instance, detection of changes in class priors [144] may be combined with ranking approaches for noise identification [71] by only modifying the noise detection threshold. This approach can easily adapt to changes in class priors, without high computational effort (no new noise model is necessary).

Evolving noise and stationary data stream Although the underlying data generation process is stationary, the noise pattern can change over time. This may occur, for instance, due to an increase in noise ratio due to the fading of a sensor (incremental noise change), less noise introduction due to the quality enhancement in data processing (abrupt noise change) or a less experienced employee verifying labels when the expert is out on holidays (reoccurring noise change). In this case, although the observed data distribution may change, such changes are due to variations of the noise distribution over time, not changes in the stationary data stream. Within this scenario, it would be interesting to further research handling changes in the observed data distribution at noise level, rather than changing the predictive model. This requires noise dealing techniques which may adapt to changes in noise patterns, such as anomaly detection [14]. Another important issue is that a test-and-train approach is often assumed for data streams, where the labels of instances are provided after predictions, for performance verification and drift analysis. Feedback based on noise labels would be necessary to properly differentiate drifts from noise.

Evolving noise and evolving data stream This is a more challenging, yet quite common scenario [34]. Besides changes in the data generation process, noise patterns also evolve over time. For instance, many problems involve multiple heterogeneous data streams, with the insertion of new sources and removal of old ones [152]. For these problems, new stream sources may have different noise patterns, as well as different dynamics. Generally, it would be very difficult to distinguish and react to noise fluctuations and real drifts. Quickly reacting to changes in the observed data could be a misleading choice due to overreaction to noise. However, designing a noise robust system may lead to underreactions to data drifts. Both cases will show a degradation in performance: in the former, due to lack of robustness to noise and in the latter, due to poor stability to handle drifts. Investigating ways to combine robustness and stability while being flexible and maintaining context awareness to data change is an interesting research direction [34].

Figure 13 depicts a simulation of a data stream of 100,000 data points, with a gradual drift starting from the 50,000th instance. The stream is generated using MOA[8], following the procedure described in Street and Kim [118]. We have added two types of noise: in the original stream concept and in the gradual drifting mechanism, from 0% (no noise) to 50%, with steps of 5%. In this figure, each plot corresponds to a different noise ratio in the main stream and lines in each plot correspond to different noise ratio in the drifting mechanism. The plots show the prequential error rate, with a window size of 1000 instances, of incremental Naïve Bayes. Analyzing the graphs, it is clear that the noise in the main data stream consistently degrades performance. (The prequential error rate goes down as the noise

[8] https://moa.cms.waikato.ac.nz.

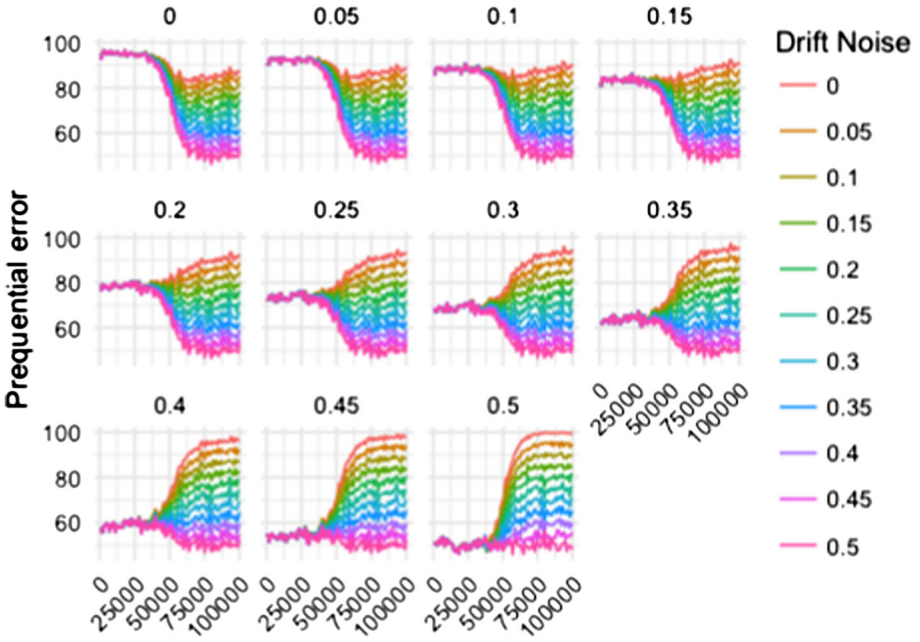

Fig. 13 Simulation of a noise stream with noise drifting concept

rate in the main stream increases.) Furthermore, it is also clear that a noise in the drifting pattern also contributes to degrading performance: the higher the noise rate related to the drift, the lower the prequential accuracy when compared to the noise-free drift.

4 Emerging topics and challenges in noisy non-binary classification

In the previous section, we have presented and formalized the noise occurrence in non-binary classification domains. In this section, we introduce the current open challenges for the same topics, stressing the areas where more attention should be given.

4.1 Issues in multiclass noise classification

Although some studies consider multiclass noise problems, different multiclass noise patterns impose numerous challenges, some of them infrequently addressed in the literature. Even state-of-the-art methods for dealing with binary class noise present considerable variation in performance when considering different multiclass noise patterns at the same noise ratio. Despite this, these issued are seldom considered in the literature. Some of aspects of this topic that could be studied further include the following:

– Would these different types of noise patterns pose the same or different challenges when dealing with multiple class noise?
– Which one would be more difficult to tackle?
– Which aspects of the problem would be more affected by considering different noise multiclass pattern?

– How do existing methods behave considering these different noise patterns?

One interesting topic for further research is how to extend methods, originally developed only for binary class, to the multiclass case. Some data transformation approaches for transforming multiclass to binary problems, e.g., One-versus-One (OVO) or One-versus-ALL (OVA), could be applied [112]. However, research on this topic generally involves random noise completely at random (NCAR), with uniform class noise distribution. Investigating how these approaches are affected by different noise patterns is an interesting topic for research. For instance, when applying a filter using a OVA decomposition, does the order in which class noise is removed matter? If so, is this influence stronger for different noise distribution among the classes?

Another open-ended problem is the relationship with imbalanced classification [100] and multiclass noise. It is reported in the literature that noise in minority classes is more harmful than in majority classes [126]. However, multiclass imbalance [128] has further issues to consider, as multiple predominant or infrequent classes may occur. It is unclear what learning difficulties multiclass noise can cause under highly imbalanced class distributions, and how to handle it effectively is an open-ended issue. Furthermore, different noise patterns can change the observed class ratio, which may influence the behavior of class imbalance techniques. Uniform class noise, for instance, may mask the observed class ratio of multiple rare classes even for low noise levels. Default class may also introduce an artificial predominant class, thus generating an artificial imbalanced problem due to the presence of noise. Possible ways to handle noise in imbalanced problems include cost-sensitive noise handling [148,151], attributing and the development of class ratio aware filtering approaches [114] considering the multiclass context.

4.2 Issues in noise multilabel classification

There are many issues that should be considered when dealing with multilabel noise, which are not properly addressed by using data transformation. A first thing to consider is that an instance may be "partially noise," in the sense that the presence or absence of some labels for the instance may be due to noise, but other labels are correct. In this context, filter-based techniques need to balance the trade-offs between partial noisy/non-noisy label identification for the instance. On the one hand, completely filtering out a partial noise instance might cause the correct labels of that instance to be lost. On the other hand, leaving an instance with partial noisy labels might negatively impact the correct classification of those labels. A binary relevance-like transformation could be used to deal with label noise individually. However, this approach would restrict the application of algorithms based on binary relevance and ignores label correlations.

Several research questions need to be further investigated regarding noise multilabel learning and label dependence:

– Is the performance of multilabel classifiers designed to incorporate known label dependence affected by the presence of noise?
– Could noise considerably affect approaches that infer label dependencies from data?
– If the artificial noise generation process does not consider label correlation, are the models unbiased with respect to the dependencies present in data without the artificial noise?
– What happens if the correlations among labels in the true data generation process are different from the correlations among noise labeling?
– Could label dependence be used to develop new techniques for dealing with noise multilabel efficiently?

Another topic that could be further studied is the influence of noisy instances in problems of label distribution learning Geng [43], Xing et al. [132], Gao et al. [39]. Label distribution learning is an emerging paradigm which includes both single-label and multilabel as special cases and aims to learn a distribution over the possible labels representing the degree to which labels describe the instance. Few studies considers noisy labeled instances for label distribution learning. In Chen and Käamäaräainen [17] exploits the reliability of labeled images to incorporate noise information in label distribution learning for age estimation. In He et al. [49], actual labels are replaced by an estimated label distribution to handle label ambiguity. The problem of learning with incomplete supervised information in the context of multilabel learning was studied in Xu and Zhou [135].

4.3 Issues in multitask problems

Although it is possible to deal within noise in each task individually, multitask applications present interesting topics for further research regarding noise handling with related tasks. How noise affects multitasks approaches, specially if the individual tasks have different noise patterns, is an interesting research topic. Studies regarding inter-task noise patterns are also interesting. In [106], structural covariance noise among regression tasks is investigated. It would be interesting to develop related studies with multitask classification problems.

Developing multitask approaches to deal with noise is also interesting. Methods that work as data preprocessors, such as filters or noise reparation techniques, could be enhanced by evaluating noisy instances together. Another possible approach is to investigate ways to transfer/adapt knowledge for dealing with noise from one domain to another. In Xiao et al. [131], an approach based on convolution neural networks for transferring noise information from a small manually cleaned noise-free data set to a large, noisy training set, in the context of image annotation was proposed. Similar approaches and applications in other domains are interesting topics for further research.

4.4 Issues in multi-instance learning

While multi-instance learning embraces the concept of negative examples within a bag at the time of creating a model, it would be interesting to develop or extend noise filtering techniques which work as preprocessors in order to evaluate whether preprocessing noisy instances within each bag may improve performance of multi-instance learning algorithms. For instance, developing filters for removing instances with a high likelihood of being negative from positive bags may help in the learning phase.

Although these approaches can be used to deal with noisy instances within each bag, dealing with bags with noise labels can be more difficult. As positive bags may have both positive and negative instances, and the individual labels or ratios of positive/negative instances within the bags are unknown, dealing with noise bags can be very challenging [77], specially if instance correlation inside the bag may influence noise occurrence [21]. A bag filtering algorithm, for instance, should evaluate all instances within a bag before deciding whether it should or should not be removed. Other approaches may include the conversion of bags into single instances [52]. A possible drawback of this approach would be the lack of data, as the size of the data set is reduced to the number of bags in the data set.

4.5 Issues in ordinal classes

Dealing with noise in ordinal classification has been mostly explored in monotonic classification, where the noise violations are very important, as some monotonic classification learning algorithms require a completely monotone training set [47]. Relabeling noise monotonic instances is a technique commonly used in the literature to deal with monotonic noise [22,29,105]. However, relabeling may not be a good option for some data sets, and other noise dealing techniques could be considered. At this point though, there is no available preprocessing technique for dealing with noisy instances in monotonic or ordinal classification, thus constituting an important open issue.

4.6 Data stream and non-stationary environment issues

Some approaches have been used to deal with noise in data streams. For instance, ensemble learning has been used to improve stability of class noise in data streams [85,142]. Other approaches are to use one-class classifiers [59] and instance selection/active learning [74, 153] to identify possible outliers in data streams. A warning state was also used to increase robustness to noise [9]. In this state, instances that differ from data distribution are firstly tagged as noise, and drift adaption only takes place if the warning state changes to out-of-control due to an increase in different instances. However, these approaches often assume that only the data stream evolves and noise is stationary, and further studies involving evolving noise patterns are welcome.

Another topic for further research is related to evolution in input and class sets. Some applications require the use of new features [79] or the appearance of new classes [78] as the stream flows. New features may be created to differentiate or select from data chunks [46, 80], with implications in noise patterns that depend on the input space. In some stream applications, the set of possible classes is not fixed and new classes may appear [122]. Distinguishing between a new class and noise require more research, especially in evolving noise scenarios.

Multiclass [23] and multilabel [102] data streams inherit the same issues from their non-stationary counterparts, with possible non-stationary class noise. For instance, in multiclass settings better preprocessing techniques may reduce noise correlation between blocks of classes while label dependencies may vary in time.

5 Concluding remarks

An important issue in classification is learning from data with class noise, as noisy classes may decrease predictive performance and difficult model induction. This problem has been addressed in the literature, and many different techniques to address class noise were proposed in the literature, both from a theoretical and practical point of view. These techniques include noise robust/tolerant techniques and data cleaning methods.

However, most noise dealing techniques have mainly been developed and validated considering binary class problems. We have shown that, even in simple synthetically generated problems in multiclass classification the performance of some state-of-the-art noise treatment methods varies considerably considering different noise patterns and skewed class distributions. We insist that these issues have not been properly discussed in the literature.

This paper also considers the class noise problem in the context of nonstandard classification problems. We examine the class noise under the multiclass, multilabel, multitask, multi-instance, ordinal and data stream classification problems. Although research involving class noise has been conducted within these frameworks, they have received considerably less attention than class noise handling in the binary classification case.

For each of these classification problems, we analyze current trends, as well as point out open-ended questions and future research directions. We also consider particular points of interest for artificial noise modeling in these classification scenarios. They are important for avoiding pitfalls and considering suitable scenarios in simulated artificial noise experimentations with more adherent and contrastive variations.

We believe that our paper can serve as a motivation to expand class noise research in other classification problems beyond binary classification problems, by pointing new topics for developing new techniques and studies considering the challenges and particularities of these non-binary classification problems.

As a final remark, it is important to stress that even though there are more studies considering binary class noise models than non-binary cases, many challenges in noise binary classes remain unsolved and require further investigation. For instance, it is known that not all class noisy instances affect the learning algorithms equally [88]. Parameter tuning [138] and classifier evaluation [109] under class noise are difficult tasks to perform. With the objective of cleaning noise in mind, further studies on the use of hybrid methods to reduce incorrect filtering, besides noise filtering and noise cleaning approaches, are necessary.

Acknowledgements This work have been partially supported by the São Paulo State (Brazil) research council FAPESP under project 2015/20606-6, the Spanish Ministry of Science and Technology under project TIN2014-57251-P and the Andalusian Research Plan under project P12-TIC-2958.

References

1. Abellán J, Masegosa AR (2010) Bagging decision trees on data sets with classification noise. In: International symposium on foundations of information and knowledge systems. Springer, pp 248–265
2. Amores J (2013) Multiple instance classification: review, taxonomy and comparative study. Artif Intell 201:81–105
3. Angluin D, Laird P (1988) Learning from noisy examples. Mach Learn 2(4):343–370
4. Baranauskas JA (2015) The number of classes as a source for instability of decision tree algorithms in high dimensional datasets. Artif Intell Rev 43(2):301–310
5. Bartlett PL, Jordan MI, McAuliffe JD (2006) Convexity, classification, and risk bounds. J Am Stat Assoc 101(473):138–156
6. Beigman E, Klebanov BB (2009) Learning with annotation noise. In: Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP: volume 1–volume 1, ACL '09, pp 280–287
7. Ben-David A, Sterling L, Tran T (2009) Adding monotonicity to learning algorithms may impair their accuracy. Expert Syst Appl 36(3):6627–6634
8. Bi Y, Jeske DR (2010) The efficiency of logistic regression compared to normal discriminant analysis under class-conditional classification noise. J Multivar Anal 101(7):1622–1637
9. Bouchachia A (2011) Fuzzy classification in dynamic environments. Soft Comput 15(5):1009–1022
10. Brefeld U, Scheffer T (2004) Co-Em support vector learning. In: International conference on machine learning (ICML), p 16
11. Breve FA, Zhao L, Quiles MG (2015) Particle competition and cooperation for semi-supervised learning with label noise. Neurocomputing 160:63–72
12. Brodley CE, Friedl MA (1999) Identifying mislabeled training data. J Artif Intell Res 11:131–167
13. Caruana R (1997) Multitask learning. Mach Learn 28(1):41–75
14. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41(3):15

15. Chapelle O, Shivaswamy P, Vadrevu S, Weinberger K, Zhang Y, Tseng B (2010) Multi-task learning for boosting with application to web search ranking. In: ACM SIGKDD international conference on knowledge discovery and data mining (KDD). ACM, pp 1189–1198
16. Charte F, Rivera AJ, del Jesús MJ, Herrera F (2015) Addressing imbalance in multilabel classification: measures and random resampling algorithms. Neurocomputing 163:3–16
17. Chen K, Kämäräinen J-K (2016) Learning with ambiguous label distribution for apparent age estimation. In: Asian conference on computer vision. Springer, pp 330–343
18. Chen P-Y, Chen C-C, Yang C-H, Chang S-M, Lee K-J (2017) milr: Multiple-instance logistic regression with lasso penalty. R J 9(1):446–457
19. Cheng W, Hüllermeier E, Dembczynski KJ (2010) Bayes optimal multilabel classification via probabilistic classifier chains. In: International conference on machine learning (ICML), pp 279–286
20. Cheplygina V, Tax DM, Loog M (2015) Multiple instance learning with bag dissimilarities. Pattern Recognit 48(1):264–275
21. Chevaleyre Y, Zucker J-D (2000) Noise-tolerant rule induction from multi-instance data. In: ICML 2000, workshop on attribute-value and relational learning
22. Daniels HA, Velikova MV (2006) Derivation of monotone decision models from noisy data. IEEE Trans Syst Man Cybern C 36(5):705–710
23. de Faria ER, de Leon Ferreira ACP, Gama J et al (2016) Minas: multiclass learning algorithm for novelty detection in data streams. Data Min Knowl Discov 30(3):640–680
24. Dembczyński K, Waegeman W, Cheng W, Hüllermeier E (2012) On label dependence and loss minimization in multi-label classification. Mach Learn 88(1–2):5–45
25. Dietterich TG, Bakiri G (1995) Solving multiclass learning problems via error-correcting output codes. J Artif Intell Res 2:263–286
26. Ditzler G, Roveri M, Alippi C, Polikar R (2015) Learning in nonstationary environments: a survey. IEEE Comput Intell Mag 10(4):12–25
27. Du J, Cai Z (2015) Modelling class noise with symmetric and asymmetric distributions. In: AAAI conference on artificial intelligence (AAAI), pp 2589–2595
28. Evgeniou T, Micchelli CA, Pontil M (2005) Learning multiple tasks with kernel methods. J Mach Learn Res 6:615–637
29. Feelders A (2010) Monotone relabeling in ordinal classification. In: IEEE international conference on data mining (ICDM). IEEE, pp 803–808
30. Frénay B, Verleysen M (2014) Classification in the presence of label noise: a survey. IEEE Trans Neural Netw Learn Syst 25(5):845–869
31. Friedman JH (1989) Regularized discriminant analysis. J Am Stat Assoc 84(405):165–175
32. Gaba A, Winkler RL (1992) Implications of errors in survey data: a Bayesian model. Manag Sci 38(7):913–925
33. Gaber MM, Gama J, Krishnaswamy S, Gomes JB, Stahl F (2014) Data stream mining in ubiquitous environments: state-of-the-art and current directions. Wiley Interdiscip Rev Data Min Knowl Discov 4(2):116–138
34. Gaber MM, Zaslavsky A, Krishnaswamy S (2005) Mining data streams: a review. ACM Sigmod Record 34(2):18–26
35. Galimberti G, Soffritti G, Maso MD et al (2012) Classification trees for ordinal responses in r: the rpartscore package. J Stat Softw 47(10):1
36. Gama J, Zliobaite I, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. ACM Comput Surv 46(4):44:1–44:37
37. Gamberger D, Boskovic R, Lavrac N, Groselj C (1999) Experiments with noise filtering in a medical domain. In: International conference on machine learning (ICML). Morgan Kaufmann Publishers, pp 143–151
38. Gamberger D, Lavrač N, Džeroski S (1996) Noise elimination in inductive concept learning: a case study in medical diagnosis. In: International workshop on algorithmic learning theory (ALT). Springer, pp 199–212
39. Gao B-B, Xing C, Xie C-W, Wu J, Geng X (2017) Deep label distribution learning with label ambiguity. IEEE Trans Image Process 26(6):2825–2838
40. Gao J, Fan W, Han J (2007) On appropriate assumptions to mine data streams: analysis and practice. In: IEEE international conference on data mining (ICDM). IEEE, pp 143–152
41. García S, Luengo J, Herrera F (2015) Data preprocessing in data mining. Springer, Berlin
42. Garofalakis M, Gehrke J, Rastogi R (2016) Data stream management: processing high-speed data streams. Springer, Berlin
43. Geng X (2016) Label distribution learning. IEEE Trans Knowl Data Eng 28(7):1734–1748

44. Ghosh A, Manwani N, Sastry P (2015) Making risk minimization tolerant to label noise. Neurocomputing 160:93–107
45. Gibaja E, Ventura S (2015) A tutorial on multilabel learning. ACM Comput Surv 47(3):52
46. Gomes JB, Gaber MM, Sousa PA, Menasalvas E (2014) Mining recurring concepts in a dynamic feature space. IEEE Trans Neural Netw Learn Syst 25(1):95–110
47. Gutiér rez PA, García S (2016) Current prospects on ordinal and monotonic classification. Prog AI 5(3):171–179
48. Gutiérrez PA, Perez-Ortiz M, Sanchez-Monedero J, Fernández-Navarro F, Hervas-Martinez C (2016) Ordinal regression methods: survey and experimental study. IEEE Trans Knowl Data Eng 28(1):127–146
49. He Z, Li X, Zhang Z, Wu F, Geng X, Zhang Y, Yang M-H, Zhuang Y (2017) Data-dependent label distribution learning for age estimation. IEEE Trans Image Process 26(8):3846–3858
50. Hernández-González J, Inza I, Lozano JA (2016) Weak supervision and other non-standard classification problems: a taxonomy. Pattern Recognit Lett 69:49–55
51. Herrera F, Charte F, Rivera AJ, del Jesus MJ (2016) Multilabel classification: problem analysis, metrics and techniques. Springer, Berlin
52. Herrera F, Ventura S, Bello R, Cornelis C, Zafra A, Sánchez-Tarragó D, Vluymans S (2016) Multiple instance learning: foundations and algorithms. Springer, Berlin
53. Hornung R (2017) Ordinal forests. Technical report 212. University of Munich, Department of Statistics
54. Hu Q, Che X, Zhang L, Zhang D, Guo M, Yu D (2012) Rank entropy-based decision trees for monotonic classification. IEEE Trans Knowl Data Eng 24(11):2052–2064
55. Ipeirotis PG, Provost F, Sheng VS, Wang J (2014) Repeated labeling using multiple noisy labelers. Data Min Knowl Discov 28(2):402–441
56. Jabbari S, Holte RC, Zilles S (2012) Pac-learning with general class noise models. In: Annual conference on artificial intelligence. Springer, pp 73–84
57. Josse J, Wager S (2016) Bootstrap-based regularization for low-rank matrix estimation. J Mach Learn Res 17(1):4227–4255
58. Khardon R, Wachman G (2007) Noise tolerant variants of the perceptron algorithm. J Mach Learn Res 8:227–248
59. Krawczyk B, Woźniak M (2015) One-class classifiers with incremental learning and forgetting for data streams with concept drift. Soft Comput 19(12):3387–3400
60. Kubat M (2015) Similarities: nearest neighbor classifiers. In: An introduction to machine learning. Springer, pp 43–64
61. Lachenbruch PA (1979) Note on initial misclassification effects on the quadratic discriminant function. Technometrics 21(1):129–132
62. Lawrence ND, Schölkopf B (2001) Estimating a kernel fisher discriminant in the presence of label noise. In: International conference on machine learning (ICML), pp 306–313
63. Leisch F, Weingessel A, Hornik K (1998) On the generation of correlated artificial binary data. SFB Adaptive information systems and modelling in economics and management science, 13. Working paper series, WU Vienna University of Economics and Business, Vienna
64. Leung T, Song Y, Zhang J (2011) Handling label noise in video classification via multiple instance learning. In: IEEE international conference on computer vision (ICCV). IEEE, pp 2056–2063
65. Li S-T, Chen C-C (2015) A regularized monotonic fuzzy support vector machine model for data mining with prior knowledge. IEEE Trans Fuzzy Syst 23(5):1713–1727
66. Li W, Vasconcelos N (2015) Multiple instance learning for soft bags via top instances. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 4277–4285
67. Li Y, Tax DMJ, Duin RPW, Loog M (2013) Multiple-instance learning as a classifier combining problem. Pattern Recognit 46(3):865–874. https://doi.org/10.1016/j.patcog.2012.08.018
68. Lin H-T, Li L (2012) Reduction from cost-sensitive ordinal ranking to weighted binary classification. Neural Comput 24(5):1329–1367
69. Little RJ, Rubin DB (2002) Statistical analysis with missing data. Wiley, New York
70. Liu B (2015) Sentiment analysis: mining opinions, sentiments, and emotions. Cambridge University Press, Cambridge
71. Lorena AC, Garcia L PF, de Carvalho ACPLF (2015) Adapting noise filters for ranking. In: Brazilian conference on intelligent systems (BRACIS), pp 299–304
72. Luengo J, Shim S-O, Alshomrani S, Altalhi A, Herrera F (2018) CNC-NOS: class noise cleaning by ensemble filtering and noise scoring. Knowl Based Syst 140:27–49
73. Ma L, Destercke S, Wang Y (2016) Online active learning of decision trees with evidential data. Pattern Recognit 52:33–45
74. Maloof MA, Michalski RS (2000) Selecting examples for partial memory learning. Mach Learn 41(1):27–52

75. Manwani N, Sastry P (2013) Noise tolerance under risk minimization. IEEE Trans Cybern 43(3):1146–1151
76. Maron O (1998) Learning from ambiguity. PhD thesis, Massachusetts Institute of Technology
77. Maron O, Lozano-Pérez T (1998) A framework for multiple-instance learning. Adv Neural Inf Process Syst 10:570–576
78. Masud M, Gao J, Khan L, Han J, Thuraisingham BM (2011) Classification and novel class detection in concept-drifting data streams under time constraints. IEEE Trans Knowl Data Eng 23(6):859–874
79. Masud MM, Chen Q, Gao J, Khan L, Han J, Thuraisingham B (2010) Classification and novel class detection of data streams in a dynamic feature space. In: European conference on machine learning and principles and practice of knowledge discovery (ECML/PKDD). Springer, pp 337–352
80. Masud MM, Chen Q, Khan L, Aggarwal CC, Gao J, Han J, Srivastava A, Oza NC (2013) Classification and adaptive novel class detection of feature-evolving data streams. IEEE Trans Knowl Data Eng 25(7):1484–1497
81. McLachlan G (1972) Asymptotic results for discriminant analysis when the initial samples are misclassified. Technometrics 14(2):415–422
82. Miao Q, Cao Y, Xia G, Gong M, Liu J, Song J (2016) Rboost: label noise-robust boosting algorithm based on a nonconvex loss function and the numerically stable base learners. IEEE Trans Neural Netw Learn Syst 27(11):2216–2228
83. Michalek JE, Tripathi RC (1980) The effect of errors in diagnosis and measurement on the estimation of the probability of an event. J Am Stat Assoc 75(371):713–721
84. Milstein I, David AB, Potharst R (2013) Generating noisy monotone ordinal datasets. Artif Intell Rev 3(1):p30
85. Minku LL, White AP, Yao X (2010) The impact of diversity on online ensemble learning in the presence of concept drift. IEEE Trans Knowl Data Eng 22(5):730–742
86. Miranda ALB, Garcia LPF, Carvalho ACPLF, Lorena AC (2009) Use of classification algorithms in noise detection and elimination. In: Corchado E, Wu X, Oja E, Herrero Á, Baruque B (eds) Proceedings of the hybrid artificial intelligence systems: 4th international conference, HAIS 2009, Salamanca, Spain. Springer, Berlin, pp 424–471
87. Montañes E, Senge R, Barranquero J, Quevedo JR, del Coz JJ, Hüllermeier E (2014) Dependent binary relevance models for multi-label classification. Pattern Recognit 47(3):1494–1508
88. Napierała K, Stefanowski J, Wilk S (2010) Learning from imbalanced data in presence of noisy and borderline examples. In: International conference on rough sets and current trends in computing. Springer, pp 158–167
89. Natarajan N, Dhillon IS, Ravikumar PK, Tewari A (2013) Learning with noisy labels. In: Advances in neural information processing systems (NIPS), pp 1196–1204
90. Nettleton DF, Orriols-Puig A, Fornells A (2010) A study of the effect of different types of noise on the precision of supervised learning techniques. Artif Intell Rev 33(4):275–306
91. Nicholson B, Sheng VS, Zhang J (2016) Label noise correction and application in crowdsourcing. Expert Syst Appl 66:149–162
92. Nowak S, Rüger S (2010) How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In: International conference on multimedia information retrieval (ICMR). ACM, pp 557–566
93. Okamoto S, Yugami N (2003) Effects of domain characteristics on instance-based learning algorithms. Theor Comput Sci 298(1):207–233
94. Ozuysal M, Calonder M, Lepetit V, Fua P (2010) Fast keypoint recognition using random ferns. IEEE Trans Pattern Anal Mach Intell 32(3):448–461
95. Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
96. Pathak D, Shelhamer E, Long J, Darrell T (2015) Fully convolutional multi-class multiple instance learning. In: International conference on learning representations (ICLR) workshop. arXiv:1412.7144
97. Pearl J (1988) Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, Burlington
98. Pérez CJ, González-Torre FJG, Martín J, Ruiz M, Rojano C (2007) Misclassified multinomial data: a Bayesian approach. RACSAM 101(1):71–80
99. Perez PS, Nozawa SR, Macedo AA, Baranauskas JA (2016) Windowing improvements towards more comprehensible models. Knowl Based Syst 92:9–22
100. Prati RC, Batista GEAPA, Silva DF (2015) Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. Knowl Inf Syst 45(1):247–270
101. Qi Z, Yang M, Zhang ZM, Zhang Z (2012) Mining noisy tagging from multi-label space. In: ACM international conference on information and knowledge management (CIKM). ACM, pp 1925–1929

102. Qu W, Zhang Y, Zhu J, Qiu Q (2009) Mining multi-label concept-drifting data streams using dynamic classifier ensemble. In: Asian conference on machine learning (ACML). Springer, pp 308–321
103. Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):81–106
104. Quinlan JR (1993) C4. 5: programs for machine learning. Elsevier, New York
105. Rademaker M, De Baets B, De Meyer H (2012) Optimal monotone relabelling of partially non-monotone ordinal data. Optim Methods Softw 27(1):17–31
106. Rakitsch B, Lippert C, Borgwardt K, Stegle O (2013) It is all in the noise: efficient multi-task Gaussian process inference with structured residuals. In: Advances in neural information processing systems (NIPS), pp 1466–1474
107. Ralaivola L, Denis F, Magnan CN (2006) CN = CPCN. In: Proceedings of the 23rd international conference on Machine learning. ACM, pp 721–728
108. Read J, Pfahringer B, Holmes G, Frank E (2011) Classifier chains for multi-label classification. Mach Learn 85(3):333–359
109. Rider AK, Johnson RA, Davis DA, Hoens TR, Chawla NV (2013) Classifier evaluation with missing negative class labels. In: International symposium on intelligent data analysis. Springer, pp 380–391
110. Rolnick D, Veit A, Belongie S, Shavit N (2017) Deep learning is robust to massive label noise. arXiv preprint arXiv:1705.10694
111. Sabzevari M, Martínez-Muñoz G, Suárez A (2018) A two-stage ensemble method for the detection of class-label noise. Neurocomputing 275:2374–2383
112. Sáez JA, Galar M, Luengo J, Herrera F (2014) Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition. Knowl Inf Syst 38(1):179–206
113. Sáez JA, Galar M, Luengo J, Herrera F (2016) INFFC: an iterative class noise filter based on the fusion of classifiers with noise sensitivity control. Inform Fusion 27:19–32
114. Sáez JA, Luengo J, Stefanowski J, Herrera F (2015) Smote-ipf: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. Inf Sci 291:184–203
115. Sánchez JS, Pla F, Ferri FJ (1997) Prototype selection for the nearest neighbour rule through proximity graphs. Pattern Recognit Lett 18(6):507–513
116. Scott C (2015) A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In: International conference on artificial intelligence and statistics (AISTATS), pp 838–846
117. Sluban B, Gamberger D, Lavrač N (2014) Ensemble-based noise detection: noise ranking and visual performance evaluation. Data Min Knowl Discov 28(2):265–303
118. Street WN, Kim Y (2001) A streaming ensemble algorithm (sea) for large-scale classification. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 377–382
119. Sulis E, Farías DIH, Rosso P, Patti V, Ruffo G (2016) Figurative messages and affect in twitter: differences between #irony, #sarcasm and #not. Knowl Based Syst 108:132–143
120. Sun B, Chen S, Wang J, Chen H (2016) A robust multi-class AdaBoost algorithm for mislabeled noisy data. Knowl Based Syst 102:87–102
121. Sun S (2013) A survey of multi-view machine learning. Neural Comput Appl 23(7–8):2031–2038
122. Sun Y, Tang K, Minku LL, Wang S, Yao X (2016) Online ensemble learning of data streams with gradually evolved classes. IEEE Trans Knowl Data Eng 28(6):1532–1545
123. Tan M, Shi Q, van den Hengel A, Shen C, Gao J, Hu F, Zhang Z (2015) Learning graph structure for multi-label image classification via clique generation. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 4100–4109
124. Teng C-M (1999) Correcting noisy data. In: Proceedings of the sixteenth international conference on machine learning. Morgan Kaufmann Publishers, San Francisco, CA, USA, pp 239–248
125. Tu H-H, Lin H-T (2010) One-sided support vector regression for multiclass cost-sensitive classification. In: International conference on machine learning (ICML), pp 1095–1102
126. Van Hulse J, Khoshgoftaar T (2009) Knowledge discovery from imbalanced and noisy data. Data Knowl Eng 68(12):1513–1542
127. Vens C, Struyf J, Schietgat L, Džeroski S, Blockeel H (2008) Decision trees for hierarchical multi-label classification. Mach Learn 73(2):185–214
128. Wang S, Yao X (2012) Multiclass imbalance problems: analysis and potential solutions. IEEE Trans Syst Man Cybern B 42(4):1119–1130
129. Wei Y, Zheng Y, Yang Q (2016) Transfer knowledge between cities. In: ACM SIGKDD conference on knowledge discovery and data mining (KDD). ACM, pp 1905–1914
130. Xiao H, Xiao H, Eckert C (2012) Adversarial label flips attack on support vector machines. In: Proceedings of the 20th european conference on artificial intelligence. IOS Press, pp 870–875

131. Xiao T, Xia T, Yang Y, Huang C, Wang X (2015) Learning from massive noisy labeled data for image classification. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 2691–2699
132. Xing C, Geng X, Xue H (2016) Logistic boosting regression for label distribution learning, In: 'Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4489–4497
133. Xu K, Liao SS, Li J, Song Y (2011) Mining comparative opinions from customer reviews for competitive intelligence. Decis Support Syst 50(4):743–754
134. Xu L, Wang Z, Shen Z, Wang Y, Chen E (2014) Learning low-rank label correlations for multi-label classification with missing labels. In: International conference on data mining (ICDM). IEEE, pp 1067–1072
135. Xu M, Zhou Z-H (2017) Incomplete label distribution learning. In: Proceedings of the 26th international joint conference on artificial intelligence. AAAI Press, pp 3175–3181
136. Xu X, Li B (2007) Multiple class multiple-instance learning and its application to image categorization. Int J Image Graph 7(03):427–444
137. Yang C-Y, Wang J-J, Chou J-J, Lian F-L (2015) Confirming robustness of fuzzy support vector machine via ξ-α bound. Neurocomputing 162:256–266
138. Yogatama D, Mann G (2014) Efficient transfer learning method for automatic hyperparameter tuning, In: Artificial intelligence and statistics, pp 1077–1085
139. Yuan X-T, Liu X, Yan S (2012) Visual classification with multitask joint sparse representation. IEEE Trans Image Process 21(10):4349–4360
140. Zeng X, Martinez T (2008) Using decision trees and soft labeling to filter mislabeled data. J Intell Syst 17(4):331–354
141. Zhang C, Wu C, Blanzieri E, Zhou Y, Wang Y, Du W, Liang Y (2009) Methods for labeling error detection in microarrays based on the effect of data perturbation on the regression model. Bioinformatics 25(20):2708–2714
142. Zhang P, Zhu X, Shi Y, Guo L, Wu X (2011) Robust ensemble learning for mining noisy data streams. Decis Support Syst 50(2):469–479
143. Zhang W, Rekaya R, Bertrand K (2006) A method for predicting disease subtypes in presence of misclassification among training samples using gene expression: application to human breast cancer. Bioinformatics 22(3):317–325
144. Zhang Z, Zhou J (2010) Transfer estimation of evolving class priors in data stream classification. Pattern Recognit 43(9):3151–3161
145. Zhou J, Liu J, Narayan VA, Ye J, Initiative ADN et al (2013) Modeling disease progression via multi-task learning. Neuroimage 78:233–248
146. Zhou Z-H, Zhang M-L, Huang S-J, Li Y-F (2012) Multi-instance multi-label learning. Artif Intell 176(1):2291–2320
147. Zhu X, Wu X (2004a) Class noise vs. attribute noise: a quantitative study. Artif Intell Rev 22(3):177–210
148. Zhu X, Wu X (2004b) Cost-guided class noise handling for effective cost-sensitive learning. In: IEEE international conference on data mining (ICDM), IEEE, pp 297–304
149. Zhu X, Wu X, Chen Q (2003) Eliminating class noise in large datasets. In: International conference on machine learning (ICML), vol 3, pp 920–927
150. Zhu X, Wu X, Chen Q (2006) Bridging local and global data cleansing: Identifying class noise in large, distributed data datasets. Data Min Knowl Discov 12(2–3):275–308
151. Zhu X, Wu X, Khoshgoftaar TM, Shi Y (2007) An empirical study of the noise impact on cost-sensitive learning. In: International joint conference on artificial intelligence (IJCAI), vol 7, pp 1168–1173
152. Zhu Y, Shasha D (2002) Statstream: statistical monitoring of thousands of data streams in real time. In: International conference on very large data bases (VLDB), VLDB Endowment, pp 358–369
153. Žliobaitė I, Bifet A, Pfahringer B, Holmes G (2014) Active learning with drifting streaming data. IEEE Trans Neural Netw Learn Syst 25(1):27–39

Ronaldo C. Prati received the M.Sc. and Ph.D. degrees in Computer Science and from the University of São Paulo, São Carlos (SP), Brazil, in 2003 and 2006, respectively. He currently is an Associate Professor in the Center of Computer Science, Mathematics and Cognition at the Federal University of ABC, Santo André (SP), Brazil. He was a visiting researcher at University of Bristol (UK) and University of Granada (Spain) in 2004 and 2017, respectively. His research interests include machine learning and data science.

Julián Luengo received the M.S. degree in computer science and the Ph.D. from the University of Granada, Granada, Spain, in 2006 and 2011, respectively. He currently acts as an Assistant Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada, Spain. His research interests include machine learning and data mining, data preparation in knowledge discovery and data mining, missing values, noisy data, data complexity and fuzzy systems.

Francisco Herrera (SM'2015) received his M.Sc. in Mathematics in 1988 and Ph.D. in Mathematics in 1991, both from the University of Granada, Spain. He is currently a Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada, Spain. He has been the supervisor of 42 Ph.D. students. He has published more than 400 journal papers, receiving more than 61.000 citations (Scholar Google, H-index 123). He is co-author of the books "Genetic Fuzzy Systems" (World Scientific, 2001), "Data Preprocessing in Data Mining" (Springer, 2015), "The 2-tuple Linguistic Model. Computing with Words in Decision Making" (Springer, 2015), Multilabel Classification. Problem analysis, metrics and techniques" (Springer, 2016), among others. He acts as Editor in Chief of the journals "Information Fusion" (Elsevier) and Progress in Artificial Intelligence (Springer), and editorial member of a dozen of journals. He is an ECCAI Fellow 2009 and IFSA Fellow 2013. He has been selected as a Highly Cited Researcher http://highlycited.com/ (in the fields of Computer Science and Engineering, respectively, 2014 to present, Clarivate Analytics). His current research interests include among others, Computational Intelligence (including fuzzy modeling, computing with words, and evolutionary algorithms), information fusion and decision making and data science (including data preprocessing, machine learning, deep learning and big data).