Journal of Systems Engineering and Electronics
Vol. 34, No. 1, February 2023, pp.36 – 46

A review of addressing class noise problems of remote sensing classification

FENG Wei [1,2,3,*], LONG Yijun [1,2,3], WANG Shuo [1,2,3], and QUAN Yinghui [1,2,3]

1. School of Electronic Engineering, Xidian University, Xi'an 710071, China;
2. Xi'an Key Laboratory of Advanced Remote Sensing, Xi'an 710071, China;
3. Key Laboratory of Collaborative Intelligence Systems, Ministry of Education, Xidian University, Xi'an 710071, China

Abstract: The development of image classification is one of the most important research topics in remote sensing. The prediction accuracy depends not only on the appropriate choice of the machine learning method but also on the quality of the training datasets. However, real-world data is not perfect and often suffers from noise. This paper gives an overview of noise filtering methods. Firstly, the types of noise and the consequences of class noise on machine learning are presented. Secondly, class noise handling methods at both the data level and the algorithm level are introduced. Then ensemble-based class noise handling methods including class noise removal, correction, and noise robust ensemble learners are presented. Finally, a summary of existing data-cleaning techniques is given.

Keywords: class noise, label noise, mislabeled classification, ensemble learning, remote sensing.

DOI: 10.23919/JSEE.2023.000034

1. Introduction

With the fast development of remote sensing techniques, abundant information can be contained. Such information has proved to be useful in applications like land cover mapping, military field, precision agriculture, and environmental modeling and monitoring [1,2]. Classification is a key issue in all the above applications and has received significant attention [3]. According to the difference in the number of requirements for labeling information, classifying methodologies can be divided into three types: unsupervised, supervised, and semi-supervised classifications [3]. Among these methods, supervised and semi-supervised methods need labeled samples to build specific learning models. Supervised learning obtains better results and a wider application field than other methods [4]. However, its performance depends strongly on the quantity and quality of the labeled samples [1,2,4]. Semi-supervised methods extract information from both the labeled and the unlabeled instances [3,4]. It not only is susceptible to noise in the original sample but also has a high risk of generating more artificial noise [3].

Real-world data is not perfect and often suffers from noise [5,6]. Labeling training instances is a costly and rather subjective task that usually induces some labeling errors in the training set [7−9]. Moreover, non-expert annotators might mislabel images due to a lack of knowledge. The presence of noisy data always produces several negative consequences for classification technologies [10−12]. Learning from noisy data can create overfitting by altering the relationship between the informative features and the measure outputs [13]. Moreover, effective noise handling is one of the most difficult problems in machine learning [7]. Therefore, how to reduce noise consequences and form an efficient training set is a major issue in both supervised and semi-supervised classification [14].

2. Types and consequences of class noise

The quality of the training data is influenced by several factors, but the class labels and attribute values are two major components directly affecting the performance of a classification algorithm [5]. These two elements are the focal spot and determining factors of designing a noise-handling method [10]. Therefore, before exploring specific research, it is significant to understand three factors: (i) the meaning of class noise and attribute noise; (ii) why

Manuscript received September 06, 2022.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (62201438;61772397;12005169), the Basic Research Program of Natural Sciences of Shaanxi Province (2021JC-23), Yulin Science and Technology Bureau Science and Technology Development Special Project (CXY-2020-094), Shaanxi Forestry Science and Technology Innovation Key Project (SXLK2022-02-8), and the Project of Shaanxi Federation of Social Sciences (2022HZ1759).

handling class noise is more crucial than addressing attribute noise; and (iii) the potential negative outcomes of class noise [5,7,15].

2.1 Types of class noise

Class noise, which is also known as label noise or mislabeled data, arises when an example is inaccurately labeled due to various reasons, such as subjectivity during the labeling process, data entry errors, or inadequacy of the information used to distinguish each example. There are three types of class noise [11,16]: contradictory examples, misclassification examples, and outliers. The contradictory samples refer to the same examples that appear more than once and are labeled with different class labels [17]. Misclassification examples, on the other hand, are examples labeled with class labels that do not match their actual labels [18].

In addition, mislabeled instances may be outliers if their labels have a low probability of occurrence in their vicinity [19]. In some cases, instances can appear unusually to the class that corresponds to their incorrect label. As a result, many techniques used to address class noise have similarities to outlier and anomaly detection techniques. Several methods, which have been developed to handle outliers and anomalies, can also be utilized for class noise.

Nonetheless, it is crucial to note that mislabeled instances do not necessarily qualify as outliers or anomalies [20]. For instance, if labeling errors occur in a boundary region where all classes are equally likely, the mislabeled instances neither appear unusually nor are rare events. Similarly, an outlier is not always a mislabeled example since it can result from attribute noise or simply be a low-probability occurrence [20].

2.2 Class noise versus attribute noise

Attribute noise occurs in the attribute values of the training set. It contains unknown attribute values, erroneous attribute values, attribute value missing, and incomplete attribute values [11]. In addition, class noise is more harmful than attribute noise when a classification model is built [21]. For example, Saez et al. reckoned that although detecting and handling noise from attribute information is the best solution for improving classification accuracy in some cases, and class noise leads to more formation of contradictory learning instances [21]. Moreover, Quinlan showed that cleaning the mislabeled training instances could result in a classifier with higher learning accuracy, but handling attribute noise of higher levels can decrease the predictive accuracy of the resulting classifier [22]. To put it another way, experimental study in [23] found that prioritizing the management of class noise over attribute noise can lead to more significant improvements in the performance of classification models. This suggests that removing attribute noise is not always necessary for building an effective model [23]. Moreover, the greater impact of class noise on the model's performance can be attributed to the facts that there are numerous features, but only one class label and that the significance of each feature for learning varies, while labels always have a significant impact [20].

2.3 Consequences of class noise

In real-world datasets, class noise is pervasive and can have adverse effects on classification models [9].

(i) The existence of class noise has been shown to result in reduced classification performance, as evidenced by theoretical proofs for basic models like *K*-nearest-neighbors (KNN) [24] and linear or quadratic classifiers [20].

(ii) Supervised classifiers, such as ensemble classifiers, may fail to function correctly with class noise. High levels of class noise can make it more difficult to learn through multiple models, as some samples become more challenging for all models, leading to poor classification by an individual model [20,25].

(iii) The presence of class noise can have various impacts on the learning process, such as increasing the required number of training instances, making the learned models more complex, increasing the number of nodes in decision trees, increasing the number of support vectors in support vector machines (SVMs) [20], and increasing the size of an ensemble (i.e., the number of base classifiers).

(iv) Class noise can make it more difficult to identify important features, affect the estimated error rate in multiclass problems, and lead to overfitting [5,13,26].

Hence, this work focuses on the review of the class noise addressing methods in the following sections.

3. Class noise handling methods

High quality labeled data set is an important factor in building a high quality learning system. Mining from noisy labeled data has been an important subfield of classification research. Several literature present an explanation of how the constraining of the class noise uses both data pre-processing-based and model optimizing-based approaches: preprocessing training set by removing noisy samples, correcting the labels of misclassified instances, as well as designing noise robust learning techniques [15,16]. The first two class noise handling methods can be briefly described by Fig. 1.

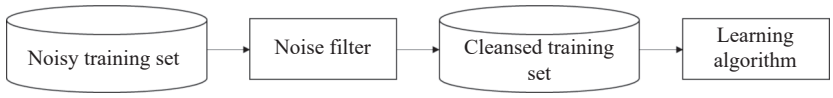


Fig. 1 General procedure for learning under class noise with training data cleansing [27]

3.1 Dealing with class noise at data level

3.1.1 Class noise identification and removal

Problems corrupted by class noise are very complex. It is difficult to achieve accurate solutions without designing specialized techniques, particularly for noise-sensitive learning methods [11]. Noise filters are commonly applied to improve classification performance. Those methods that rely on preprocessing techniques identify and remove noisy instances from the training set [8,28]. Moreover, the efficiency of a noise filter can be affected by several factors, including the attributes of the training data, the degree of class noise, the noise tolerance of the supervised classifier, and the generalization ability of the learning algorithms [28].

A simple and comment data cleansing method is to remove the mislabeled instances. Many such methods exist in the class noise-related literature. For example, classification filtering uses the predictions of some high noise robustness classifiers to identify the potentially mislabeled samples [20]. Thongkam et al. built an SVM model with all the training samples, then removed all misclassified instances from the training set [29]. The filtering approach not only induces relatively low computational costs but also is easy to implement [30]. However, a major drawback of this approach is that some valuable instances might be dropped from the data set [31]. And the operation is particularly harmful to small sample learning and class imbalance learning. Although the noise filtering approach cannot completely clean up the mislabeled instances caused by a human operator or measurement errors when the noise levels are greater than 30% [32], keeping those samples may hinder performance more than removing too many correctly labeled samples [8].

Hughes et al. extended the above method by deleting the labels of misclassified instances that were deemed unreliable by experts and then used semi-supervised learning with both labeled and unlabeled data [33]. This approach preserves the distribution of the data, thereby avoiding a potential bias in the results [20]. Nonetheless, this approach encounters a conundrum known as the chicken-and-egg problem, whereby learning in a noisy setting can result in suboptimal classifiers, despite the need for accurate classifiers for effective classification filtering.

Noisy data removal can also be said to be selecting potentially correct data. According to Wang et al., the identification and removal of noisy data can aid in the training of neural networks and subsequently improve their ability to detect clean instances [34]. Therefore, based on this idea, Wang et al. proposed a framework combining a network training algorithm with a scalable regularized regression (SRR) approach. The training algorithm involves both supervised and semi-supervised training and randomly selects one training scheme for each mini-batch with a predefined probability. Following the completion of training, the signal to noise ratio (SNR) approach receives features and labels of data and a noisy set and then employs theoretical guarantees to identify noisy data. To accomplish this, Wang et al. proposed an equivalent leave-one-out test approach as a penalized linear model, in which non-zero mean-shift parameters can serve as an indicator for noisy data [34].

The blame-based noise reduction (BBNR) algorithm is an easy-to-understand technique for removing noisy instances from a dataset. This algorithm eliminates all samples that are responsible for the misclassification of their nearest neighbors, as well as those that can be safely removed without causing any other instance to be misclassified. Notably, BBNR does not require a highly accurate classifier for noise filtering, but it may incur increased computational complexity [35].

3.1.2 Class noise identification and correction

All the above algorithms primarily aim to enhance the data quality by identifying and eliminating class noise. Teng demonstrated that using a classifier constructed from corrected data can lead to greater predictive power than using filtered data [31]. Teng [31] introduced an alternative approach called polishing, whereby instead of removing the noisy instances, they are repaired by replacing their erroneous class label values with predicted class labels. And the corrected instances are then reinserted into the dataset. The depuration algorithm presented in [32] employs an iterative approach to adjust the class labels of instances whose assigned label conflicts with the majority of their neighboring instances' labels.

Cluster analysis is a viable option for identifying mislabeled instances by leveraging neighborhood consistency, as demonstrated by the depuration algorithm [32]. An instance with a label that is inconsistent and does not align with the labels of its adjacent clusters is potentially mislabeled [20]. In [36], this neighborhood consistency criterion is utilized to create a clustering-based algorithm that rectifies the labels of mislabeled instances. For instance, the density-based spatial clustering of applications with noise (DBSCAN) algorithm employs two parameters, i.e., radius of the cluster (eps) and the minimum number of points in the cluster, to partition all points into three categories: core points, border points, and noise points. Specifically, all points take themselves as the point of circles of the radius eps and the number of other points they enclose is counted; on the ground of such a number, every point can be categorized [37]. This method can effectively distinguish isolated samples and thus label them as noise points [38].

Nevertheless, noise correction is only practical for small datasets since it tends to be time-consuming [11]. Despite some studies indicating that complete or partial noise correction in the training data, while leaving the test data still under noise, can boost test performance results over no preprocessing at all [11,31], this method can introduce additional noise (due to correction failure) into the training data if too many genuinely clean examples are mislabeled [39].

3.2 Dealing with class noise at classifier level

3.2.1 Robustness of learning algorithms

Robustness refers to an algorithm's capacity to generate models that are resilient to data corruption and less susceptible to the effects of noise. A more robust classification algorithm builds classifiers that are less influenced by noise, resulting in more comparable models built from both clean and noisy data [11]. In the presence of noisy data, robustness takes precedence over performance results since it provides insight into the anticipated behavior of a learning method against noise, especially when the attributes of the noise are uncertain. [11]. By considering the robustness in algorithm designs, it may be more feasible to find real patterns in different contexts in reality [40]. Therefore, it is reasonable to discuss robustness against noise.

3.2.2 Robust algorithms against noise

Class noise can have a significant impact on decision trees, leading to instability in their construction. However, they are still suitable for ensemble methods. To improve the robustness of decision trees, an appropriate splitting criterion should be carefully selected. In the presence of class noise, various node split criteria were compared in [41]. The imprecise information-gain, based on imprecise probabilities and uncertainty measures, was found to improve accuracy compared to the Gini index, the information gain ratio, and the information gain. Post-pruning is another approach commonly used to deal with noise in decision trees [41]. This technique involves reducing the size of decision trees by removing sections of the tree that provide little power to classify instances, which reduces the complexity of the final classifier and improves predictive accuracy by reducing overfitting caused by the over-specialization over the isolated (and usually noisy) examples [11]. Nonetheless, according to Gamberger et al., this method is less efficient than noise elimination and correction [7]. It is worth noting that even a robust learner may have poor performance if the noise level is relatively high [11].

Li et al. introduced a new method for robust representation learning and noise handling called selective-supervised contrastive learning (Sel-CL) [42]. Sel-CL is an extension of supervised contrastive learning (Sup-CL), which is a powerful technique for representation learning but can be negatively impacted by noisy labels. Sel-CL addresses the main issue with Sup-CL, in which the creation of noisy pairs mislead representation learning due to the pair-wise nature of the method [42]. To alleviate this problem, confident pairs are selected out of noisy ones for Sup-CL without knowing noise rates. It is achieved by first identifying confident examples via measuring the agreement between the learned representations and given labels. Then, confident pairs are built using the confident examples, and more confident pairs are identified from the representation similarity distribution in the built confident pairs. These obtained confident pairs are then used for Sup-CL to improve representations. The method's robustness was evaluated on multiple noisy datasets, and the experiments showed that it outperformed state-of-the-art methods [42].

The KNN classifiers [43] are highly sensitive to class noise, especially when the neighborhood's size is small [44]. In the presence of noise, it is often necessary to preprocess the data to obtain good performance. Saez et al. suggested data complexity measures to predict in advance when a noise filter would statistically improve the prediction results of 1NN [28].

4. Ensemble-based class noise handling methods

4.1 Ensemble methods for class noise filtering

4.1.1 Ensemble-based class noise removal

The removal of mislabeled instances is an important preprocessing step in classification, but traditional methods face the challenge of potentially removing valuable data along with the mislabeled ones. To overcome this limitation, ensemble approaches are widely used, where multiple base classifiers are trained on the same dataset, and their votes are combined to make predictions [8,13,18,45−48]. Ensemble-based filtering methods attempt to improve the quality of the training data by detecting and removing mislabeled instances based on the votes of the base classifiers [45]. The majority vote filter and the consensus filter are two typical approaches for ensemble-based filtering [8]. The majority vote filter eliminates instances incorrectly classified by over half of the base classifiers. In addition, the consensus filter is too strict as it requires that all base classifiers fail to classify an instance, and may eliminate only a small portion of the mislabeled instances [8,45]. However, a majority vote filter does not only remove mislabeled instances but also all the clean training instances that the ensemble classifier wrongly classified. The filter cannot differentiate these false positives from mislabeled instances (true positives), which is a significant drawback. This is because clean training instances that were incorrectly identified as noise contain crucial information, such as boundary instances that are vital for classifier design [45]. Hence, neither of these approaches is completely effective for mislabeled instance filtering [45].

Verbaeten et al. investigated the issue of mislabeled training examples in classification by applying ensemble methods (bagging and boosting) to preprocess the training set [13]. Their study utilized C4.5 as a base classifier [49]. Two distinct approaches they introduced are

(i) Filtering based on voting (consensus vote and majority vote) of base classifiers of a bagging ensemble.

(ii) Filtering based on removing training examples that obtained high weights in the boosting process. Indeed, mislabeled examples are assumed to have high weights.

Results indicated that majority vote filters were more effective than consensus filters, and bagging-majority vote filters outperformed boosting filters, which tended to incorrectly remove many accurately labeled instances with high weights.

Zhu et al. put forward a technique to detect and eliminate mislabeled instances in extensive or distributed datasets by dividing them into subsets [18]. Their approach involves partitioning a large dataset E into subsets and learning a set of classification rules R_i for each subset of E. A special rule set GR_i is then selected from R_i and used to evaluate all instances in the original dataset. The approach involves utilizing two error count variables, namely, the local error count and the global error count. To identify noise, these variables keep track of how each instance in E performs with the good rule sets generated from all the subsets. Typically, exceptions do not trigger GR_i and noise is more likely to invalidate GR_i, resulting in a greater likelihood of noisy instances receiving significant error values as opposed to clean examples [18]. Noise is identified using two schemes: majority and non-objection, and then removed along with a portion of good examples after each round. The procedure can be repeated if the filtering result is unsatisfactory. This method was proved to be effective for extensive datasets.

Miranda et al. combined four classifiers trained using distinct machine learning techniques to form a heterogeneous ensemble and used voting to identify mislabeled instances [50]. The detected noise was subsequently eliminated, resulting in increased accuracy. However, this method removes instances that fall on the incorrect side of the classification boundary, which can be problematic [20,51]. Additionally, multiple parameters must be taken into account while selecting various techniques as base classifiers for a given dataset.

Feng et al. proposed a cleaning technique called the ensemble method based on the noise detection metric (ENDM) to clean corrupted training sets [10]. First, an ensemble classifier is trained and used to derive four metrics assessing the likelihood of a sample being mislabeled. Three thresholds are set for each metric to maximize the classifying performance on a corrupted validation dataset when using three different ensemble classifiers, namely Bagging, AdaBoost, and KNN. These thresholds are used to identify and then either remove or correct the corrupted samples [10].

Edge analysis is another method to detect mislabeled instances [52]. The edge of an instance is defined as the sum of the weights of weak classifiers that misclassify the instance in a boosting ensemble [53], in contrast to the ensemble margin proposed by Schapire et al. [54]. Instances with high edge values are often misclassified by weak classifiers, indicating low confidence. This approach involves classifying harder observations correctly in later rounds by initially classifying correctly

labeled observations incorrectly. It removes instances with top edge values, typically 5%, as mislabeled instances tend to have high edge values due to persistent misclassification [20].

In outlier removal boosting (ORBoost) [55], data cleansing is integrated with the learning process instead of being conducted after learning. During the boosting process, instance weights above a certain threshold are set to zero. This method is more robust than Adaboost because it pays less attention to class noise. However, its performance is good only when the noise level is low, and the threshold selection is sensitive and requires validation set tuning [20]. The problem of selecting an appropriate ensemble-based class noise filter remains a major issue in ensemble learning. Sluban et al. conducted a study to examine how ensemble diversity affects the performance of class noise detection. They hypothesized that ensemble diversity helps identify noise detection ensembles that perform well [56]. The study analyzed the majority and consensus ensemble voting schemes and found that increasing diversity in ensembles using majority voting did not improve the noise detection performance and may even degrade it. Conversely, for heterogeneous ensembles utilizing consensus voting for noise detection, higher diversity resulted in higher precision in class noise detection.

4.1.2 Ensemble-based class noise correction

As discussed in Subsection 3.1, the process of noise removal can lead to the elimination of valuable information, and, in certain situations, noise correction has proven to produce more favorable outcomes than simply discarding the noise from the dataset [31]. Rebbapragada et al. proposed the utilization of active learning as a means to address class noise issues [39]. To identify mislabeled data, they proposed two scores: active label correction (ALC)-mislabeled and ALC-disagreement. The ALC-mislabeled score assesses the likelihood of an example x being mislabeled by calculating the difference in probabilities between the existing and predicted labels. The higher the score, the greater the probability of mislabeling x. The instances are arranged in descending order based on their scores, and the k highest scoring examples are submitted for expert evaluation. On the other hand, ALC-disagreement chooses examples for relabeling those not explicitly mislabeled ones. Instead, it selects examples that demonstrate a considerable level of ambiguity in their predicted labels and can, therefore, be viewed as "hard-to-classify" examples. This confusion is expressed by the probability distribution over the class labels: the closer it is to a uniform distribution, the more uncertain the classification [39]. Finally, the predicted class labels that receive the most votes are used to update the mislabeled examples. Two automated cleaning techniques, namely single-pass discarding and correcting, are used for comparison with ALC-disagreement. Single-pass discarding removes instances from the dataset if their probability or committee votes on the current label are lower than the probabilities or votes on the predicted label. In contrast, single-pass correcting is a simple technique that corrects misclassified examples by updating them to their predicted labels. The findings revealed that active learning performs better than these two automated data cleaning methods above. Nonetheless, similar to any active learning strategy, human expertise is necessary, which is a significant drawback compared to automated class noise handling.

Miranda et al. expanded their noise detection approach described in Subsection 4.1 to correct mislabeled data [50]. Instances identified as noise are reclassified based on the classes that are predicted most by the noise detection classifiers. As a comparison, the authors also proposed a hybrid technique where KNN is used to decide whether to remove or correct data identified as class noise. The results demonstrated that the classifiers constructed using both class noise handling methods can achieve higher accuracy than those using the original training set. Furthermore, it was found that the classification accuracy obtained through noise correction and hybrid methods was comparable for the majority of the datasets, despite their expectation that the hybrid approach would perform better. Additionally, Shao et al. grouped data by KNN for each class and divided them into subsets, which were fed into ensemble branches [57]. Subsequently, these classification models of the ensemble branches can yield graphs that represent local data manifolds, and correction suggestions for a final correct label result were obtained using the information of the original sample-label pairs and sample-correction pairs. To make the correction more convincing, it also attempts to measure the confidence of the result which can optimize the training process for the next epoch. However, both methods were found to be less effective than their noise removal technique described in Subsection 4.1. Moreover, as discussed in Subsection 4.1, their noise detection algorithm tends to identify many important correctly classified samples as noise. Thus, an imprecise noise identification method can lead to less effective noise removal and correction, regardless of how reasonable the noise identification strategy is. Table 1 is a brief summary of methods mentioned above.

Table 1 Class handling method: removal and correction

Description	Noise removal	Noise correction
Basic procedure	Detecting and eliminating noisy data through noise filters	Detecting and correcting the labels of the noisy instances with predicted labels
Non-ensemble method	SVM for outlier detection [29]	Polishing: identify noisy data and replace its label by predicted labels [31]
	Semi-supervised learning of probabilistic models for noise removal [33]	Clustering: detect and group noisy data based on a neighborhood consistency constraint [32,36]
	Scalable penalized regression for noise detection [34]	
	The blame-based noise reduction algorithm [35]	Density-based spatial clustering of applications with noise [37]
Ensemble method	Bagging-majority vote filters [13]	Active learning: active label correction (ALC-mislabeled and ALC-disagreement) to identify mislabeled data [39]
	Identifying and eliminating data through subsets and error counts [18]	
	Heterogeneous ensemble with four different base classifiers [50]	Single-pass discarding and correcting [39]
	Ensemble method based on the noise detection metric [10]	
	Edge analysis: a boosting ensemble to detect noisy data based on the sum of the weights of weak classifiers [53]	Classification: relabel the noisy data by the class that is most predicted [50]
	Outlier removal boosting [55]	

4.1.3 Exploiting the ensemble margin for class noise filtering

The use of ensemble margins has been proposed as a means of designing noise filters. In [45], noisy instances are identified as those that are either mislabeled in the training data, or are intrinsically ambiguous and challenging to categorize because their label value conflicts with the majority of the other instance label values, despite having similar attribute values. Guo [45] proposed an algorithm that uses the unsupervised ensemble margin to detect class noise, which refers to instances that are misclassified by most of the base classifiers in the ensemble.

In other words, class noise is identified as instances that are classified with high margins as belonging to the wrong class. Guo's algorithm sorts all misclassified training instances in descending order based on their unsupervised margin values, and removes a portion of the highest margin examples that are classified incorrectly. The algorithm then employs two noise removal strategies, namely adaptive filtering and fixed filtering, to estimate or confirm the amount of noise. The results indicated that the ensemble margin can effectively identify class noise, and the adaptive filtering strategy is advantageous in cases where the amount of noise is uncertain while in other cases, they are comparable.

In [58], a reverse boosting algorithm is introduced. This method distinguishes between safe, noisy, and borderline patterns, and assigns them different weights during boosting. The weights of safe patterns are increased, those of noisy patterns are decreased, and those of borderline patterns remain unchanged. The samples are classified into these three categories using a committee machine called parallel perceptron. The ensemble margin of the parallel perceptron is used to classify samples into three categories: safe, noisy, and borderline. The proposed approach enhances the performance of the parallel perceptron algorithm when dealing with datasets that contain noisy labels. However, classical perceptron generally outperforms reverse boosting [20].

4.2 Class noise tolerant ensemble learners

One of the widely used techniques for creating ensembles is AdaBoost [59], which is preferred due to its simplicity and flexibility [60]. However, AdaBoost tends to overfit class noise because it assigns large weights to mislabeled instances during the later stages of training [20]. To address this issue, various methods have been proposed to update the weights more cautiously to decrease the susceptibility of boosting to class noise [20,61]. For example, the AveBoost2 [62] algorithm replaces the weight ω_i^{t+1} of the instance at step $t+1$ by the following expression:

$$\frac{tw_i^{(t)} + w_i^{(t+1)}}{t+1}.$$

AveBoost2 achieves larger training errors but smaller generalization errors than AdaBoost. Besides, AveBoost2 slows down the growth of misclassified instance weights, making it more robust to class noise than AdaBoost. Similarly, modified AdaBoost (MadaBoost) [63] limits the instance weight to the initial probability, preventing weights from becoming excessively large as they do in AdaBoost. This method has been shown not to overfit on noisy data. Averaged boosting (A-Boost) [64] differs from AdaBoost in that it calculates weights based on the error rate of the current hypothesis on the original training examples and uses the average of the product of the base hypotheses and weights, while AdaBoost uses the

sum. On noisy data, A-Boost performs similarly to bagging. However, modifications of weights, which are common losses in machine learning, may not always be effective, especially when dealing with high levels of noise [65].

Cao et al. introduced a new boosting technique called noise detection based Adaboost (ND-Adaboost) [60]. In their work, They conducted an analysis of class noise detection-based loss function and ensemble margin, and subsequently proposed a novel loss function. The proposed approach is an extension of Adaboost that integrates the class noise-detection-based loss function. This is done to adjust the weight distribution at each iteration and to control the ensemble training error bound via a regeneration condition [60].

In a separate study, Krieger et al. [66] proposed two approaches to mitigate the impact of class noise in boosting. The first approach involves limiting the number of iterations of Adaboost to prevent overfitting, but the authors did not investigate effective methods to determine the optimal number of iterations. The second approach is to smooth the boosted classifier by combining bagging and boosting as follows:

(i) This hybrid method creates K bootstrapped training subsets comprising a certain p percentage of the original training set;

(ii) K boosted classifiers are trained for a specified number M of iterations;

(iii) The K predictions are aggregated to form the final prediction. This approach aims to enhance the diversity of the boosted classifier, ultimately leading to improved performance in noisy environments compared to Adaboost [66].

Bagging is a method that is more effective than boosting in the noisy environment. Bagging improves the diversity of base classifiers by creating different subsets of training sets through bootstrap sampling, thereby reducing the impact of each mislabeled sample on the classifier [20]. Abellan et al. found that bagging ensembles of credal decision trees, which are based on imprecise probabilities and information-based uncertainty measures, were effective in classification issues with high levels of noise in the class variable [41]. In a comparative empirical study [67], it was shown that bagging-C4.5 outperformed in the majority of datasets with 0 to 20% noise, but failed when 30% noise was introduced.

Research indicates that the choice of sampling size for bagging may not have a significant impact on the ensemble's generalization performance. However, the optimal size of bootstrap samples is likely to vary depending on the specific application, particularly in the presence of class noise. As a result, subsampling is a promising avenue to explore [68]. Sabzevari et al. demonstrated that bagging, which involved training unpruned decision trees on bootstrap samples ranging from 10% and 40% of the original training set size, was more resistant to class noise than standard bagging, which relied on a sampling ratio of 100% of the original data [68].

In the context of machine learning, problems involving multiple classes can become increasingly complex and may lead to higher chances of incorrect classifications, particularly in the presence of noise [11]. Several studies have shown that one way to mitigate this issue is to break down the multiclass problem into several binary sub-problems [69]. This method involves a two-step process:

(i) Problem division: The problem is divided into several binary sub-problems with each sub-problem solved by independent binary classifiers.

(ii) Combination of the outputs: The One-vs-One (OVO) decomposition strategy is employed [69], which divides a classification problem with M classes into $M(M-1)/2$ binary subproblems. For each subproblem, a classifier is trained only on the training examples corresponding to the pair of classes (λ_i, λ_j), notably, $i < j$.

In [69], the effectiveness of the OVO decomposition strategy in improving the accuracy of baseline classifiers in the presence of class noise was evaluated. The robust learners C4.5 and repeated incremental pruning to produce error reduction (RIPPER) [70] robust learners and the noise-sensitive KNN method were tested with and without the usage of OVO. The results showed that the OVO decomposition enhanced the accuracy of all the baseline classifiers in noisy datasets, possibly due to the distribution of noisy examples in different subproblems and the combination of information from different classifiers [69].

5. Conclusions

Class noise is a complex problem in remote sensing due to a significant number of mislabeled instances in training data, which can negatively impact classification outcomes. There exist various methods to address class noise, such as noise removal, noise correction, and class noise-robust methods. However, there is no isolated method that is entirely effective for all noisy data, and machine learning practitioners must sort out the most relevant method for their field of application. For example, class noise-robust methods may be sufficient if the class noise is only marginal. In some cases, removing mislabeled instances can be more effective than correcting them, despite the fact that many data cleansing methods are efficient and straightforward to implement. Nevertheless, instance removal methods may remove too many

correct instances, leading to over-cleansing, which is an important problem for imbalanced datasets in remote sensing while retaining mislabeled instances has much poorer performance than removing too many correctly labeled samples. It is essential to find a compromise between retaining mislabeled instances that can potentially worsen classification results and removing too many correctly labeled samples in the future research.

The use of ensemble learners, particularly random forests, has shown to be more resilient against mislabeling in comparison to single classifiers. Indeed, ensemble-based methods for handling class noise are based on specific assumptions. Firstly, data cleansing methods rely on various heuristics to distinguish mislabeled instances from exceptions, which reflect different definitions of class noise. Secondly, class noise-robust methods assume that avoiding overfitting is enough to deal with class noise. As a result, ensemble-based methods strike a balance between using instances as they are and identifying potentially mislabeled instances, contributing to their success in handling class noise.

There remain many unanswered research questions concerning class noise, and numerous areas are yet to be explored. One potential avenue is the use of semi-supervised learning, which has the benefit of not altering the instance distribution. It would be worthwhile to investigate whether this approach could be more effective in managing class noise than merely eliminating questionable instances from noisy data. Another possibility is to use multiclass decomposition to alter the distribution of noisy examples in sub-problems, which can enhance class separability. This method can also be used for noise detection and data selection. Ensemble margin is also a promising approach for designing classifiers against noise and identifying noisy data. Recent studies have revealed a correlation between the generalization performance of an ensemble classifier and the distribution of margins on training examples. Additionally, the random forest has been widely demonstrated to be the most resilient method against noise in ensemble learning, making it a worthwhile research direction to explore how noise filtering performs in random forest classification.

References

[1] FENG W, HUANG W J, BAO W X. Imbalanced hyperspectral image classification with an adaptive ensemble method based on SMOTE and rotation forest with differentiated sampling rates. IEEE Geoscience and Remote Sensing Letters, 2019, 16(12): 1879–1883.

[2] FENG W, QUAN Y H, DAUPHIN G, et al. Semi-supervised rotation forest based on ensemble margin theory for the classification of hyperspectral image with limited training data. Information Sciences, 2021, 575: 611–638.

[3] FENG W, DAUPHIN G, HUANG W J, et al. Dynamic synthetic minority over-sampling technique based rotation forest for the classification of imbalanced hyperspectral data. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2019, 12(7): 2159–2169.

[4] HE F, WANG R, JIA W M. Fast semi-supervised learning with anchor graph for large hyperspectral images. Pattern Recognition Letters, 2020, 130: 319–326.

[5] ZHU X Q, WU X D. Class noise vs. attribute noise: a quantitative study. Artificial Intelligence Review, 2004, 22(3): 177–210.

[6] ALGAN G, ULUSOY I. Image classification with deep learning in the presence of noisy labels: a survey. Knowledge-Based Systems, 2021, 215: 106771.

[7] GAMBERGER D, LAVRAC N, DZEROSKI S. Noise detection and elimination in data preprocessing: experiments in medical domains. Applied Artificial Intelligence, 2000, 14(2): 205–223.

[8] CARLA E, FRIEDL M. Identifying mislabeled training data. Journal of Artificial Intelligence Research, 1999, 11(1): 131–167.

[9] FENG W, DAUPHIN G, HUANG W J, et al. New margin-based subsampling iterative technique in modified random forests for classification. Knowledge-Based Systems, 2019, 182: 104845.

[10] FENG W, QUAN Y H, DAUPHIN G. Label noise cleaning with an adaptive ensemble method based on noise detection metric. Sensors, 2020, 20(23): 6718.

[11] GARCIA S, LUENGO J, HERRERA F. Dealing with noisy data. Data Preprocessing in Data Mining, 2015, 72: 107–145.

[12] LI P L, HE X H, CHENG X J, et al. An improved categorical cross entropy for remote sensing image classification based on noisy labels. Expert Systems with Applications, 2022, 205: 117296.

[13] VERBAETEN S, ASSCHE A V. Ensemble methods for noise elimination in classification problems. Proc. of the International Workshop on Multiple Classifier Systems, 2003: 317–325.

[14] MELLOR A, BOUKIR S, HAYWOOD A, et al. Using ensemble margin to explore issues of training data imbalance and mislabeling on large area land cover classification. Proc. of the International Conference on Image Processing, 2014: 5067–5071.

[15] WANG R Y, STOREY V C, FIRTH C P. A framework for analysis of data quality research. IEEE Trans. on Knowledge and Data Engineering, 1995, 7(4): 623–640.

[16] CATAL C, ALAN O, BALKAN K. Class noise detection based on software metrics and ROC curves. Information Sciences, 2011, 181(21): 4867–4877.

[17] HERNANDEZ M A, STOLFO S J. Real-world data is dirty: data cleansing and the merge/purge problem. Data Mining and Knowledge Discovery, 1998, 2(1): 9–37.

[18] ZHU X Q, WU X B, CHEN Q J. Eliminating class noise in large datasets. Proc. of the 20th International Conference on Machine Learning, 2003: 920–927.

[19] PECHENIZKIY M, TSYMBAL A, PUURONEN S, et al. Class noise and supervised learning in medical domains: the effect of feature extraction. Proc. of the 19th IEEE International Symposium on Computer-Based Medical Systems, 2006: 708–713.

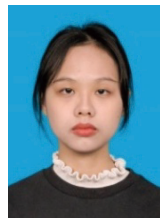
[20] FRENAY B, VERLEYSEN M. Classification in the presence of label noise: a survey. IEEE Trans. on Neural Net-

works and Learning Systems, 2013, 25(5): 845–869.
[21] SAEZ J A, CORCHADO E. ANCES: a novel method to repair attribute noise in classification problems. Pattern Recognition, 2022, 121: 108198.
[22] QUINLAN J R. Induction of decision trees. Machine Learning, 1986, 1(1): 81–106.
[23] AL-SABBAGH K W, STARON M, HEBIG R. Improving test case selection by handling class and attribute noise. Journal of Systems and Software, 2022, 183: 111093.
[24] GUO G D, WANG H, BELL D A, et al. KNN model-based approach in classification. Proc. of the OTM Confederated International Conferences on the Move to Meaningful Internet Systems, 2003: 986–996.
[25] ALI K M, PAZZANI M J. Error reduction through learning multiple descriptions. Machine Learning, 1996, 24(3): 173–202.
[26] VAN DEN HOUT A, VAN DER HEIJDEN P G M. Randomized response, statistical disclosure control and misclassification: a review. International Statistical Review, 2002, 70(2): 269–288.
[27] BURGERT T, RAVANBAKHSH M, DEMIR B. On the effects of different types of label noise in multi-label remote sensing image classification. IEEE Trans. on Geoscience and Remote Sensing, 2022, 60: 5413713.
[28] SAEZ J A, LUENGO J, HERRERA F. Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification. Pattern Recognition, 2013, 46(1): 355–364.
[29] THONGKAM J, XU G D, ZHANG Y C, et al. Support vector machine for outlier detection in breast cancer survivability prediction. Proc. of the Asia-Pacific Web Conference, 2008: 99–109.
[30] SEGATA N, BLANZIERI E, DELANY S J, et al. Noise reduction for instance-based learning with a local maximal margin approach. Journal of Intelligent Information Systems, 2010, 35(2): 301–331.
[31] TENG C M. Correcting noisy data. Proc. of the 16th International Conference on Machine Learning, 1999: 239–248.
[32] BARANDELA R, GASCA E. Decontamination of training samples for supervised pattern recognition methods. Pattern Recognition, 2000, 1876: 621–630.
[33] HUGHES N P, ROBERTS S J, TARASSENKO L. Semi-supervised learning of probabilistic models for ECG segmentation. Proc. of the 26th IEEE Annual International Conference, 2004, 1: 434–437.
[34] WANG Y K, SUN X, FU Y. Scalable penalized regression for noise detection in learning with noisy labels. Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022: 346–355.
[35] DELANY S J, CUNNINGHAM P. An analysis of case-base editing in a spam filtering system. Advances in Case-Based Reasoning, 2004, 3155: 128–141.
[36] PRASAD M N, SOWMYA A. Multi-class unsupervised classification with label correction of hrct lung images. Proc. of the International Conference on Intelligent Sensing and Information Processing, 2004: 51–56.
[37] ARAFA A, EL-FISHAWY N, BADAWY M, et al. RN-SMOTE: reduced noise SMOTE based on DBSCAN for enhancing imbalanced data classification. Journal of King Saud University-Computer and Information Sciences, 2022, 34(8): 5059–5074.
[38] BUSHRA A A, YI G M. Comparative analysis review of pioneering DBSCAN and successive density-based clustering algorithms. IEEE Access, 2021, 9: 87918–87935.
[39] REBBAPRAGADA U, BRODLEY C E, SULLA-MENASHE D, et al. Active label correction. Proc. of the IEEE 12th International Conference on Data Mining, 2012: 1080–1085.
[40] HANCOX-LI L. Robustness in machine learning explanations: does it matter? Proc. of the Conference on Fairness, Accountability, and Transparency, 2020: 640–647.
[41] ABELLAN J, MASEGOSA A R. An experimental study about simple decision trees for bagging ensemble on datasets with classification noise. Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 2009, 5590: 446–456.
[42] LI S K, XIA X B, GE S M, et al. Selective-supervised contrastive learning with noisy labels. Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022: 316–325.
[43] DUDA R O, HART P E, STORK D G. Pattern classification. Hoboken: John Wiley & Sons, 2001.
[44] OKAMOTO S N, OBUHIRO Y. An average-case analysis of the k-nearest neighbor classifier for noisy domain. Proc. of the 15th International Joint Conference on Artificial Intelligence, 1997, 1: 238–243.
[45] GUO L. Margin framework for ensemble classifiers: application to remote sensing data. Talence: University of Bordeaux, 2011.
[46] GUO L, BOUKIR S. Ensemble margin framework for image classification. Proc. of the IEEE International Conference on Image Processing, 2014: 4231–4235.
[47] SLUBAN B, GAMBERGER D, LAVRAC N. Ensemble-based noise detection: noise ranking and visual performance evaluation. Data Mining and Knowledge Discovery, 2013, 38: 265–303.
[48] KHOSHGOFTAAR T M, ZHONG S, JOSHI V. Enhancing software quality estimation using ensemble-classifier based noise filtering. Intelligent Data Analysis, 2005, 9(1): 3–27.
[49] QUINLAN J R. C4.5: programs for machine learning. San Francisco: Morgan Kaufmann Publishers Inc, 1993.
[50] MIRANDA A L B, GARCIA A C, CARVALHO L P, et al. Use of classification algorithms in noise detection and elimination. Proc. of the International Conference on Hybrid Artificial Intelligence Systems, 2009: 417–424.
[51] GUYON I, MATIC N, VAPNIK V. Discovering informative patterns and data cleaning. USAMA M F, GREGORY P S, PADHRAIC S, et al. ed. Advances in Knowledge Discovery and Data Mining. Menlo Park: American Association for Artificial Intelligence, 1996.
[52] WHEWAY V. Using boosting to detect noisy data. Proc. of the Pacific Rim International Conference on Artificial Intelligence Workshop Reader, 2001, 2112: 123–130.
[53] BREIMAN L. Arcing the edge. Berkeley: University of California, 1997.
[54] SCHAPIRE R E, FREUND Y, BARTLETT P, et al. Boosting the margin: a new explanation for the effectiveness of voting methods. The Annals of Statistics, 1998, 26(5): 1651–2080.
[55] KARMAKER A, KWEK S. A boosting approach to remove class label noise. International Journal of Hybrid Intelligent Systems, 2006, 3(3): 169–177.

[56] SLUBAN B, LAVRAC N. Relating ensemble diversity and performance: a study in class noise detection. Neurocomputing, 2015, 160: 120–131.
[57] SHAO H C, WANG H C, SU W T, et al. Ensemble learning with manifold-based data splitting for noisy label correction. IEEE Trans. on Multimedia, 2021, 24: 1127–1140.
[58] CANTADOR I, DORRONSORO J R. Boosting parallel perceptrons for label noise reduction in classification problems. Proc. of the International Work Conference on the Interplay between Natural and Artificial Computation, 2005: 586–593.
[59] FREUND Y, SCHAPIRE R E. Experiments with a new boosting algorithm. Proc. of the 13th International Conference on Machine Learning, 1996: 148–156.
[60] CAO J J, KWONG S, WANG R. A noise-detection based adaboost algorithm for mislabeled data. Pattern Recognition, 2012, 45(12): 4451–4465.
[61] VEZHNEVETS A, BARINOVA O. Avoiding boosting overfitting by removing confusing samples. Proc. of the European Conference on Machine Learning, 2007: 430–441.
[62] OZA N C. Aveboost2: boosting for noisy data. Proc. of the International Workshop on Multiple Classifier System-Multiple Classifier Systems, 2004, 3077: 31–40.
[63] DOMINGO C, WATANABE O. Madaboost: a modification of adaboost. Proc. of the 13th Annual Conference on Computational Learning Theory, 2000: 180–189.
[64] KIM Y D. Averaged boosting: a noise-robust ensemble method. Advances in Knowledge Discovery and Data Mining, 2003, 2637: 388–393.
[65] BARTLETT P L, JORDAN M I, MCAULIFFE J D. Convexity, classification, and risk bounds. Journal of the American Statistical Association, 2006, 101(473): 138–156.
[66] KRIEGER LONG C, WYNER A. Boosting noisy data. Proc. of the 18th International Conference on Machine Learning, 2001: 274–281.
[67] ABELLAN J, CASTELLANO J G, MANTAS C J. A new robust classifier on noise domains: bagging of credal C4. 5 trees. Complexity, 2017. DOI: 10.1155/2017/9023970.
[68] SABZEVARI M, MARTINEZ-MUNOZ G, SUAREZ A. Small margin ensembles can be robust to class-label noise. Neurocomputing, 2015, 160: 18–33.
[69] SAEZ J A, GALAR M, LUENGO J, et al. Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition. Knowledge and Information Systems, 2014, 38(1): 179–206.
[70] WILLIAM W C. Fast effective rule induction. Proc. of the 12th International Conference on Machine Learning, 1995: 115–123.

Biographies

FENG Wei was born in 1985. She received her B.S. degree in computer science and technology from Northeast Agricultural University, Harbin, China, in 2009, M.S. degree in computer applications technology from North Minzu University, Yinchuan, China, in 2013, and Ph.D. degree in information science and technology from Université Michel de Montaigne-Bordeaux 3, Bordeaux, France, in 2017. She worked as a postdoctoral researcher with the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China, from 2017 to 2019. She is an associate professor with the Department of Remote Sensing Science and Technology, School of Electronic Engineering, Xidian University, Xi ’an, China. Her research interests include remote sensing, machine learning, and image processing.
E-mail: wfeng@xidian.edu.cn



LONG Yijun was born in 2000. She received her B.E. degree from Xidian University. She is pursuing her M.S. degree in electronic information engineering at Xidian University, Xi ’an, China. Her research interests include deep learning and the classification of remote sensing images.
E-mail: yjlong@stu.xidian.edu.cn

WANG Shuo was born in 1997. She is currently pursuing her M.S. degree in control science and engineering with the Department of Remote Sensing Science and Technology, School of Electronic Engineering, Xidian University, Xi’an, China. Her research interests include hyperspectral target extraction and remote sensing image classification.
E-mail: shuow@stu.xidian.edu.cn

QUAN Yinghui was born in 1981. He received his B.S. and Ph.D. degrees in electrical engineering from Xidian University, Xi’an, China, in 2004 and 2012, respectively. He is currently a full professor with the Department of Remote Sensing Science and Technology, School of Electronic Engineering, Xidian University. His research interests include radar imaging, radar signal processing, and radar remote sensing.
E-mail: yhquan@mail.xidian.edu.cn