A Study on Class Noise Detection and Elimination

Luís Paulo F. Garcia
ICMC,
Universidade de São Paulo (USP)
São Carlos, Brazil
Email: lpgarcia@icmc.usp.br

Ana Carolina Lorena
CMCC,
Universidade Federal do ABC (UFABC)
Santo André, Brazil
Email: ana.lorena@ufabc.edu.br

André C. P. L. F. Carvalho
ICMC,
Universidade de São Paulo (USP)
São Carlos, Brazil
Email: andre@icmc.usp.br

Abstract—Real data may present a significant amount of noise, generated by inaccuracies in data collection, transmission and storage. The presence of noisy data in a training dataset used for the induction of a Machine Learning model may increase the training time and the complexity of the induced model, resulting in the deterioration of its predictive performance for new data. Noise may be found in the input and target attributes. In this study, we are concerned with noise in the class label of the target attribute. For such, we propose and experimentally investigate some simple class noise detection and elimination strategies for classification problems, introducing controlled noise levels in five UCI datasets originally free of inconsistencies. The results obtained in the experiments performed show the potential of the proposed approaches.

Keywords—Class Noise; Ensemble of Classifiers; Machine Learning.

I. INTRODUCTION

In a typical classification dataset, we have a set of data objects \((x_i, y_i)\), where each \(x_i\) is a tuple of \(d\) attribute values describing a given object, and \(y_i\) corresponds to the class label of \(x_i\). In this scenario, we may have two types of noise: in the predictive attributes and in the target attribute (class labels) [1]. In this study, we deal with class noise, where the inaccuracies are in the labels of some objects.

Noisy data do not follow the patterns from the domain they belong to. Noise can be caused by inaccuracies, corruption, distortion or contamination of samples [2]. Outliers, which are correct cases that correspond to exceptions, can also be considered noisy, since they do not follow the general patterns in data [3].

When analyzing datasets containing noisy data, some problems may arise. Although a large number of techniques is robust to noisy, incomplete, inconsistent and redundant data, a growing number of studies identifies problems related to low quality data [4], [5], [6]. Real data collected from data bases are estimated to contain at least 5% of noise level [7], [8].

In particular, the presence of noisy data in a training dataset used for the induction of a Machine Learning (ML) model [9] may increase the training time and the complexity of the induced model, resulting in the deterioration of its predictive performance for new data, thus influencing security and reliability when using the model in critical environments [10]. Therefore, the elimination of noisy data can lead to the induction of models of higher quality.

In a typical classification dataset, we may have two types of noise: in the predictive attributes and in the target attribute (class labels) [1]. In this study, we deal with class noise, where the inaccuracies are in the labels of some objects. Specifically, we use ensembles of classifiers to support the identification of noisy data. Inconsistencies in their predictions can indicate a potential noise. Afterwards, two strategies for dealing with the noisy objects were studied. In the first one, all potential noisy data identified were removed. Nevertheless, this removal can also harm the predictive performance of the classifiers, which are now generated from a smaller training dataset. By using the set of predictions from the ensemble of classifiers, the second strategy estimates a new class value for those objects considered noisy, preserving the original number of objects in the datasets. The set of classifiers to be used in noise detection and elimination was chosen based on how much they agree in their predictions for the training data. We obtained good results with such approach, despite its simplicity. As baseline for comparison, we used three consolidated techniques that are considered to be noise-filtering approaches according to [11].

Previous work from the literature have already reported the successful use of ML algorithms for noise pre-processing [12], [13], [14], [15], [16]. In [14] we also investigated the use of ensembles of classifiers in noise handling. There, differently from here, noisy data is continuously eliminated based on the classification performance achieved on a validation set. In this work a simpler strategy is employed and we also included more classification techniques. This work has three main contributions: (a) a simple method to choose the classifiers to be used in noise detection and elimination/relabeling, which showed good experimental results; (b) we study further the benefit of data relabeling, which is able to maintain the original number of objects in the pre-processed datasets and tries to correct their inaccuracies; (c) we performed a careful and extensive experimental analysis on datasets known to have no inconsistencies. An increasing level of controlled noise was injected into such datasets, allowing an analysis for different levels of noise. Therefore, we were able to investigate how the noise handling techniques behave for increasing levels of controlled noise.

This paper is structured as follows: in Section II we introduce the methods employed in this work, showing the techniques proposed in this paper. Section III presents the experiments performed, whose results are discussed in Section...
IV. Section V presents the conclusions.

II. NOISE HANDLING

As previously discussed, the presence of noise in datasets may significantly influence the quality of the model induced by a ML technique, affecting its processing time, complexity and predictive performance. In classification problems, noise in data can be divided into two groups: the first is related to errors in input or predictive attributes, while the second affects the classes (target attribute), through an incorrect labeling [17]. We investigated here the second type of noise.

For noise identification, we employed ensembles of classifiers. We chose a set of classification techniques from different learning paradigms, so that they can complement each other: support vector machines, $k$-nearest neighbors, decision trees, naive bayes and neural networks (Table I).

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Abbreviation</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support Vector Machines</td>
<td>SVM</td>
<td>[18]</td>
</tr>
<tr>
<td>$k$-nearest neighbor</td>
<td>$k$-NN $k=3,5,9$</td>
<td>[9]</td>
</tr>
<tr>
<td>CART</td>
<td>CART</td>
<td>[19]</td>
</tr>
<tr>
<td>C4.5</td>
<td>C4.5</td>
<td>[20]</td>
</tr>
<tr>
<td>Random Forests</td>
<td>RF</td>
<td>[21]</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>NB</td>
<td>[22]</td>
</tr>
<tr>
<td>Multilayer Perceptron</td>
<td>MLP</td>
<td>[23]</td>
</tr>
</tbody>
</table>

Given a dataset to be pre-processed, we generate classifiers using each one of the previous classification techniques. We then choose the $m$ models with more similar predictions on the training data to compose the ensemble, that is, those classifiers that agree most in their predictions. We expect that these will be the classifiers with better joint ability to identify class mislabelings.

The predictions of the $m$ selected classifiers are aggregated for noise identification. Two aggregation strategies were tested. The first was consensus, where an example is identified as noise if all $m$ classifiers predict an incorrect label for it. The second was majority voting, where an example is considered noisy if the majority of the classifiers misclassify it. This pre-processing approach is applicable on classification problems, once the outputs of the objects are considered when deciding whether they are noisy or not.

Next, for dealing with the potential noisy objects, we use two approaches: removal and reclassification. Removal eliminates the noisy data from the original dataset, generating a new reduced dataset. Reclassification, as the name suggests, reclassifies the object, assigning a new label to it, keeping the original size of the dataset. This is carried out by considering the predictions of the classifiers used in the noise detection and the original label of the object. When the majority of the predictions of the ensemble of classifiers differ from the current object label, this label is changed accordingly.

As baseline for comparison, we used three noise-filtering approaches [11]. Basically, they use the labels of the $k$ nearest neighbors of an object to define if it is noisy or not, eliminating this instance from the dataset. This procedure removes mislabeled data and borderlines [24] and tends to enlarge the margin of separation between classes and smoothing the decision border used to separate data. Borderlines are examples from different classes close to each other, since they are next to the decision border separating the classes. These examples can be considered unreliable, since even a small amount of noise may have moved them to a wrong class [25].

The Edited Nearest Neighbor (ENN) and Repeated ENN (RENN) are techniques which remove an example if the label of its $k$ nearest neighbors is different from its label [26]. While ENN takes into account the $k$ nearest neighbors only one time, the RENN applies ENN repeatedly until all objects have the majority of their neighbors with the same class. In this last case, the process of smoothing is severe and enlarges significantly the decision border. All-$k$NN [27], instead of using a fixed value of $k$, defines a range with $i = 1, ..., k$ nearest neighbors for the application of the ENN rule. After the loop is complete, those objects signaled as noisy are removed from the dataset. It should be noticed that all the baseline techniques considered remove the objects identified as unreliable.

III. EXPERIMENTS

This section describes the main aspects of the experiments performed in this study.

A. Datasets

We chose five datasets from the UCI repository [28] that contain no class inconsistencies according to their description. They are artificially generated or automatically collected. Two of them represent the results of games (Tic-Tac-Toe and Chess) and one describes a psychological experiment (Balance scale).

Table II provides a description of the main characteristics of these datasets, showing their number of examples, of attributes and majority class error (ME), which is the error rate obtained by predicting all objects as belonging to the class with most examples in the dataset (indicated within parenthesis for each dataset). In all datasets we have two classes only, because of our relabeling strategy. Originally, the balance scale dataset has three classes, but we used only two of them in the experiments (Left and Right). For the Chess dataset (King-Rook vs. King-Knight), we generated 1000 examples.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$#$ Examples</th>
<th>$#$ Attributes</th>
<th>ME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tic-Tac-Toe</td>
<td>958</td>
<td>9</td>
<td>34.6 (positive)</td>
</tr>
<tr>
<td>Monks 1</td>
<td>556</td>
<td>6</td>
<td>50.0 (0 and 1)</td>
</tr>
<tr>
<td>Monks 2</td>
<td>601</td>
<td>6</td>
<td>34.2 (0)</td>
</tr>
<tr>
<td>Chess</td>
<td>1000</td>
<td>13</td>
<td>23.0 (safe)</td>
</tr>
<tr>
<td>Balance (L and R)</td>
<td>576</td>
<td>4</td>
<td>50.0 (L and R)</td>
</tr>
</tbody>
</table>

For each dataset, we injected random noise levels, from 5% to 40%, at intervals of 5%. This dataset pollution was
controlled, such that we can assess which are the noisy objects. Moreover, since the choice of the noisy objects is random, we generated 10 different noisy versions of the datasets, for each noise level considered. Next, all datasets were sampled according to the stratified $k$-fold cross-validation methodology, with $k = 10$, forming ten pairs of training and testing datasets.

B. Methodology

We divided the experiments into three steps. In the first step, several classifiers were induced for all training datasets. The three classifiers with more similar predictions in a given training dataset are chosen for its pre-processing. Although all classifiers could be combined, we opted for using the smallest odd number of classifiers that could form an ensemble ($m = 3$). A larger number of classifiers would make consensus voting too restrictive.

The second stage, named pre-processing, is intended to detect the potential noisy instances. In this stage we used the ensembles of classifiers following the consensus and majority voting schemes (Section II) and also the baseline noise-filtering techniques. This was performed for all polluted training datasets generated. Since we have control on which were the noisy objects introduced, we were able to evaluate the precision and recall in their identification. These rates allow us to evaluate the effectiveness of the techniques in proper noise detection.

Precision is defined as the number of correctly identified noisy cases divided by the number of examples identified as noisy. A high precision technique shall be able to correctly identify as noise those cases which really correspond to noise, making fewer mistakes of nominating safe examples as noise. Recall is the number of correctly identified noisy cases divided by the total number of noise introduced. Therefore, a high recall rate indicates that many of the noise data introduced were identified, while a low recall indicates that many of the noisy cases were ignored.

Afterwards, also in the second stage, the noisy objects identified are removed or relabeled, generating pre-processed versions of the polluted training datasets. The baseline noise-filtering techniques always remove the noisy cases identified.

In a third stage, we induced classification models using the original and pre-processed training datasets and assessed their accuracy on the test data, which is free from noise. Herewith, we can evaluate whether the noise handling schemes were able to generate higher quality datasets for the induction of accurate classification models. We used Decision Trees induced by the C4.5 algorithm in this evaluation. Since the C4.5 pruning operation also intends to make the tree less prune to overfitting due to noisy data, we can access the effect of the previous noise handling techniques for a ML technique with some robustness to noise.

We induced C4.5 classification models using the original data, the polluted data and the pre-processed data. Afterwards, the performance of those classifiers is compared in order to verify if higher quality datasets could be obtained.

IV. Results

First, we analyze the results achieved in noise identification. Next, we present the accuracy results of C4.5 classifiers when using the original and pre-processed datasets.

A. Performance in Noise Identification

According to our strategy for choosing the classifiers for noise identification, different sets of classifiers are used for each particular training dataset. The classifiers used in the ensembles for noise identification are those presented in Table I. These were the classifiers combined for noise identification in both consensus and majority voting ensembles. As already described, we chose the three classifiers that agree most in their predictions on training data for composing the ensembles for each dataset. Besides being simple, this scheme showed satisfactory results in the experiments performed. The SVM and $k$-NN were the classifiers most used in the combinations, followed by C4.5. For all classification techniques we used the default parameter values of the R tool, used in the experiments, except from $k$-NN, where different values of $k$ were employed (3, 5 and 9).

Considering the ensemble techniques, we first compare the precision and recall results of consensus and majority voting. Figure 1 plots the precision vs recall results achieved by theses techniques for all datasets and noise levels considered. Each point corresponds to the precision x recall result obtained for one of the polluted datasets generated. Therefore, these results concern all datasets and noise levels. We can note that consensus voting was more conservative in detecting noisy cases, showing lower recall. On the other hand, its precision was in general high and the noise data identified were indeed noisy. Nonetheless, majority voting also showed good precision rates allied to better recall rates. In fact, majority voting points are more concentrated towards the point (1,1) of the graph, which represents an ideal point of maximum precision and recall.

Figure 2 shows a similar plot for the results of the baseline noise-filtering techniques. In this case, the performance of RENN is more concentrated towards the point (1,1) and we
then chose this technique for a further comparison with the majority voting results.

For this comparison, we first computed the $F^\beta$ measure, combining the precision and recall values of the techniques, as presented in Equation 1. We used $\beta = 0.5$, giving more importance to precision than to recall, in accordance to previous work from literature [15], [16], where the authors say that precision should be preferred in noise identification such that the noisy cases identified are indeed noise.

$$F^\beta = \left(1 + \beta^2\right) \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$ (1)

Figure 3 shows a graphical comparison of the $F^\beta$-measure values obtained by the majority voting ensemble and the RENN technique. Again, each point represents the result obtained for a polluted dataset. The X axis represents the majority voting $F^\beta$ measure, against the Y axis, which represents the $F^\beta$ measure for the RENN technique. The right inferior part of the graph represents cases for which the performance of majority voting is better than that of RENN. Almost all points are on this side of the graph, showing a superior performance of the ensemble technique.

Regarding the percentage of objects identified as noisy by the techniques, for all cases, majority voting showed percentages similar (higher or lower) to the rates of noise introduced in the datasets. Larger differences occurred for the balance dataset. RENN, on the other hand, always identified as unreliable more cases than those introduced in the datasets. This difference was more severe for the Monks 2 dataset. For instance, for 5% of noise level in this dataset, RENN signalized 44.28% objects as unreliable, while majority voting signalized 5.36% of them. It should be noticed, however, that those objects identified as noisy or unreliable by both techniques do not necessarily correspond to actual noise introduced in the datasets. To better assess whether the noisy cases were correctly identified, the precision and recall values of the techniques should be observed.

Figure 4 shows the average precision and recall of the noise identification techniques, for the different noise levels considered. Therefore, each point corresponds to the average precision or recall obtained by the ensemble or RENN technique in noise identification, for all datasets. The aim is to verify how the noise identification techniques behave for increasing noise levels. The results were averaged due to space constraints and also to allow observing the tendency of the results.

From these graphs, it is possible to see that the precision of the ensembles in noise identification was higher than that of RENN, for all noise levels. Nonetheless, the ensemble precision results are more affected by an increasing level of noise. On the other hand, the recall of RENN technique was usually higher than that of the ensembles, specially for increasing noise levels. We believe that, when the noise level becomes too high, the classifiers that compose the ensembles tend to learn the wrong patterns from noisy data. The ability to identify noise is then impaired, and many noisy cases are not identified, which reflects in a reduction in precision and recall.

**B. Classification Performance**

Figures 5 to 9 show the relative accuracy rates of C4.5 classifiers using the pre-processed versions of all datasets, when compared to the accuracy rates obtained with the noisy datasets. We present relative results using as baseline the
performance achieved for the noisy data in order to visualize how much the different techniques for noise pre-processing are able to recover from the errors introduced in the datasets. The more elevated is a bar representing the relative performance for a given pre-processing, the better is the accuracy achieved by the C4.5 classifier induced. Bars below the X axis indicate a performance deterioration.

For the Tic-tac-toe dataset, starting from 30%, these improvements start to decrease for the ensemble pre-processing techniques. Other interesting observation are the poor results of RENN technique for noise levels lower than 30%. After that rate the technique had good results.

For the Chess dataset, the ensemble techniques showed some improvement after 20% of noise level and also for noise values between 25% and 35%. On the other hand, for all noise levels, the RENN technique had significant worse results. In some cases the accuracy becomes almost 10% worse than that achieved using the noisy dataset.

For the Monks 1 dataset, the predictive accuracies improved after 15% of noise level. Before this value, there is a performance degradation and the accuracies of the classifiers for all noise handling techniques deteriorates. All techniques had good results for more than 20.

For the Monks 2 dataset, the predictive accuracy deteriorated in all cases, when compared to the accuracy values using both the original and noisy datasets. The reason for this is that precision and recall rates in noise identification for this particular dataset were, in general, relatively low.

For the Balance Scale dataset, there were some improvements of accuracy with noise pre-processing. The ensemble technique had good results until 15% of noise level. Afterwards, the RENN noise filtering technique becomes better.

The results suggest that the investigated approaches for noise handling were usually successful in generating pre-processed datasets, since the classifiers performances were in general less influenced by noise when compared to the noisy versions of the datasets. Although in some cases there were some degradation of the results, there were clear improvements for some datasets. The degradations can be attributed to cases
where safe points were considered noisy or vice-versa by the pre-processing techniques. Nonetheless, the good results, when compared to the noisy data performances, indicate a successful identification of some of the noisy points by the techniques investigated.

Finally, reclassification was slightly better than simple noise removal and RENN noise-filtering. This was expected, since the later techniques reduce the number of training data by removing the noisy instances. This influences the classifiers performance, since less data are available for the induction of the classifiers. This shows the importance of a proper pre-processing of noisy datasets for classification problems, even when simple strategies are used.

V. CONCLUSION

This study investigated and proposed some simplistic noise handling strategies for classification datasets that use ensembles of classifiers for noise identification. According to the experimental results, the consensus voting approach was too conservative and was unable to identify most of the noisy cases. A majority voting combination of the classifiers’ outputs was more successful in identifying noisy data, showing higher precision values in this identification, and obtaining classification models with better predictive performance.

When inducing classifiers using the pre-processed versions of the datasets obtained by majority voting, there were improvements of the classification performance when compared to the use of noisy data. Therefore, this technique was, in general, successful in recovering from the errors introduced in the datasets. The ensemble also performed well compared to known noise-filters from the literature. This shows the efficiency of the ensemble technique in obtaining higher quality data sets and that a noise handling strategy is a promising approach for these cases. Furthermore, trying to correct the erroneous labels by a reclassification strategy can be considered better than simply removing the noise identified.

As future work, we plan to investigate further the influence of the number of classifiers combined in the ensembles in the precision and recall obtained in noise identification. We shall also analyse the results of the noise handling techniques in more details and try to propose new strategies for choosing the classifiers to be combined for noise identification. Moreover, in this paper we only investigated datasets with two classes, for simplifying our relabeling strategy. We plan to propose and investigate strategies also suited for multiclass datasets.

ACKNOWLEDGMENTS

The authors would like to thank FAPESP (process no. 2011/14602-7), CNPq, CAPES and CeMEAI for their financial support.

REFERENCES