Pattern Recognition ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Pattern Recognition

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

# Active cleaning of label noise

Rajmadhan Ekambaram*, Sergiy Fefilatyev, Matthew Shreve, Kurt Kramer, Lawrence O. Hall, Dmitry B. Goldgof, Rangachar Kasturi

*Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620-5399, USA*

ARTICLE INFO

*Article history:*
Received 20 May 2015
Received in revised form
16 August 2015
Accepted 17 September 2015

*Keywords:*
Support vectors
Label noise
Mislabeled examples

ABSTRACT

Mislabeled examples in the training data can severely affect the performance of supervised classifiers. In this paper, we present an approach to remove any mislabeled examples in the dataset by selecting suspicious examples as targets for inspection. We show that the large margin and soft margin principles used in support vector machines (SVM) have the characteristic of capturing the mislabeled examples as support vectors. Experimental results on two character recognition datasets show that one-class and two-class SVMs are able to capture around 85% and 99% of label noise examples, respectively, as their support vectors. We propose another new method that iteratively builds two-class SVM classifiers on the non-support vector examples from the training data followed by an expert manually verifying the support vectors based on their classification score to identify any mislabeled examples. We show that this method reduces the number of examples to be reviewed, as well as providing parameter independence of this method, through experimental results on four data sets. So, by (re-)examining the labels of the selective support vectors, most noise can be removed. This can be quite advantageous when rapidly building a labeled data set.



## 1. Introduction

Mislabeled examples in the training data perturb the learning process and are likely to have an adverse effect on the accuracy of a classifier. Label noise can best be examined while an expert is available to label the data. In this paper, we present a procedure for correcting training data that contains label noise. In particular, we investigate finding mislabeled examples using support vector machines (SVM) [1–3]. This work was motivated by a search for oil-droplet particles in images from underwater platform in the aftermath of Deepwater Horizon Oil Spill. In the search for underwater oil-droplets a new class (actually suspected fish eggs) was found, but because it was a new class, examples were mislabeled. In this case, it was very important to find all "oil droplets". The presence of mislabeled examples in the training data is a critical problem and several approaches have been proposed in the literature [4–13] to address it. No approach, to our knowledge, focuses solely on the support vectors of an SVM classifier to address this problem. In our previous work [14], we hypothesized that if examples in the training data were erroneously labeled they will tend to be on the margin and get chosen as support vectors of the SVM classifier. In this work, we extend the approach to reduce the number of examples to be reviewed and provide extensive experimental results to demonstrate the validity of the hypothesis. We note that our work is not limited to images. It is also the case that we ignore noise in images, which has been dealt with in many places [15].

We did two sets of experiments to remove the label noise examples. The first set of experiments showed that around 85% and 99% of the label noise examples were selected as support vectors of one-class SVM (OCSVM) and two-class SVM (TCSVM) respectively. In these experiments we also found that large numbers of training examples (around 55% for OCSVM and between 42% and 46% for TCSVM) were selected as support vectors. This leads to reviewing more than 40% of the examples to remove 10% mislabeled or noise examples. Motivated by the results shown in [6], we rank ordered the support vectors of TCSVM examples based on their class probability. This method showed that most of the label noise examples have low probability for the class to which they are assigned. But we found three problems with this approach: (1) dependency on classifier parameters, (2) the need for the selection of the number of examples to review in each batch, and (3) the need for a threshold to stop the review process. To overcome these problems we have developed a new method

* Correspondence to: 4202 E. Fowler Avenue University of South Florida ENB-118 Tampa, FL 33620-5399 U.S.A. Tel.: +1 813 974-3652; fax: +1 813 974-5456
*E-mail addresses:* rajmadhan@mail.usf.edu (R. Ekambaram), sfefilatyev@gmail.com (S. Fefilatyev), Matthew.Shreve@parc.com, mshreve@mail.usf.edu (M. Shreve), kurt@larcosaqua.com, kurtkramer@gmail.com (K. Kramer), lohall@mail.usf.edu (L.O. Hall), goldgof@mail.usf.edu (D.B. Goldgof), r1k@mail.usf.edu (R. Kasturi).

http://dx.doi.org/10.1016/j.patcog.2015.09.020
0031-3203/

and applied it in a second set of experiments. This new method assumes that all the label noise examples are selected as support vectors of a TCSVM, and builds another noise free classifier, which is used to select the potential noise examples in the support vector examples selected in the first step. This leads to a significantly reduced number of examples to be reviewed to remove the label noise examples.

This paper shows that to correct label noise it is enough to review a subset of the support vectors of a trained TCSVM classifier. We re-labeled the noise examples in the support vectors with the help of a human expert. The validity of this approach is demonstrated on four datasets (UCI letter recognition, MNIST digit dataset, Wine quality dataset [16], and Wisconsin Breast Cancer dataset) that contain artificially introduced label-noise. The experimental results show that up to 99%, as shown in Table 6, of the incorrectly assigned labels in the training set are selected as support vectors of an SVM classifier. Using our proposed approach the number of examples to be reviewed can be drastically reduced. The paper is organized as follows. A discussion of previous work related to label noise error is presented in Section 2. The intuition behind our work and the algorithm are explained in Section 3. A detailed description of the experiments and a performance comparison with the probabilistic based method proposed in [6] are presented in Section 4. Section 5 contains our conclusions.

## 2. Related work

There are many different approaches to identify and remove mislabeled (label noise) examples that have been explored in the literature. The intuition behind a few of the methods is closely related to our work, i.e., in targeting the important examples, but differ in the criterion used to define importance. The criterion used is information gain in the work by Guyon et al. [4], distance to the separating hyperplane in the work by Rebbapragada et al. [5], and probability in the work by Rebbapragada [6], and Brodley et al. [17]. In the work by Guyon et al. [4], a method was proposed to select or reduce the number of examples instead of using all the examples for training the classifiers. The examples were manually verified after being put in decreasing order by information gain criteria to find the most important and potentially mislabeled examples. The examples which produced more information gain were more useful to the classifier, as well as more suspicious. The main idea of this method is similar to our approach. The examples were reviewed based on the information gain criteria and in our approach the criteria are implicitly defined by the large margin principle. We differ from [4] in classifier(s), how we rank examples, the strict use of human in the loop and analysis of the number of trials to remove examples and what percentage of mislabels can be found for removal. In the work by Rebbapragada et al. [5], examples were selected for labeling in an active learning framework using an SVM classifier. The unlabeled examples which lie close to the separating hyperplane were selected for labeling. The intuition of this method is very close in principle to our method, but we are different in the following: our examples are labeled and we only examine the support vector examples. The examples selected for labeling in [5] may or may not become a support vector and online training for large datasets is time consuming. The method of Rebbapragada [6] and Brodley et al. [17] have similarities to our proposed approach. They classified the training data from the classifier created using SMO in Weka [18] and generated a probability with the classification [19]. Then the examples which received low probability were verified by the labeler. The examples are not necessarily support vectors and depending on where the probability threshold for reviewing examples lies, some support vectors on the wrong side of the boundary may be ignored. We compare with this work below.

A few more methods are related to our work, but their approach is different. In the work by Gamberger et al. [7], a complexity measure was defined for the classifier and a weight was assigned to each example. The method is iterative and in each round the example with the highest weight is selected. The selected example is examined for label noise, if its weight is greater than the threshold. Our method is also iterative but the number of rounds is independent of the number of noise examples and also does not require any threshold. In the method of Brodley and Friedl [8], an automatic noise removal technique that also removes good examples was introduced. It increases the classifier accuracy, but may miss a number of mislabels which is problematic if there is a small class of interest. In the method of Zhu et al. [9], a rule based method was proposed to distinguish exceptions and mislabeled examples. The intuition behind the method in [9] is similar to the method in [8], but it can be applied for distributed, large scale datasets. The dataset was divided into subsets and rules were generated for all the subsets. Examples in each subset were classified by the rules generated from all the subsets. The assumption is that the mislabeled examples were misclassified by more rules than exceptions. We do not consider exceptions in our method, but our method can be applied independently in each location of a distributed large scale dataset as long as a sufficient number of positive and negative examples is present in each location. The method of Muhlenbach et al. [10] used geometrical structure to find the mislabeled examples. The Relative Neighborhood graph of the Toussaint method was used to construct a graph. An example is considered as bad or doubtful if its proportion of connections with examples of the same class in the graph is smaller than the global proportion of the examples belonging to its class. This method is closely related to our method, because in both methods examples which are closest to examples from other classes are suspected, but the geometry considered in this method is local whereas in our method the global position of all examples is considered at the same time. A kernel based method was proposed by Valizadegan and Tan [11] for this problem. In this method, a weighted $k$ nearest neighbors (kNN) approach was extended to a quadratic optimization problem. The expression to be optimized depends only on the similarity between the examples and hence can also be solved by projecting the attributes into higher dimensions with the help of a kernel. The examples whose labels were switched to maximize the optimization expression were considered mislabeled. This method is similar to our method in using the optimization function, but the objective of the optimization function is different. In the work by Rebbapragada and Brodley [12] and Rebbapragada et al. [13], examples are clustered pair wise and a confidence is assigned to each example using the Pair Wise Expectation Maximization (PWEM) method. The classifiers which take a confidence value as input instead of labels can make use of this information. A confidence measure can also be calculated using our method, but the criterion used is different.

The other approach to solve this problem is to mitigate the effect of the label noise examples on the classifier. In the Adaboost learning algorithm, the weights of the misclassified instances are increased and weights of correctly classified instances are decreased. This will create a group of base classifiers which correctly predict the examples that have large weights. The work of Ratsch et al. [20] and Dietterich [21] shows that AdaBoost tends to overfit in the presence of mislabeled examples. In order to avoid building base classifiers for noisy examples, a method was proposed by Cao et al. [22] to reduce the weights of the noisy examples using kNN and Expectation Maximization methods. In the work of Biggio et al. [23,24] and Niaf et al. [25], the SVM problem formulation was modified to handle the label noise

Please cite this article as: R. Ekambaram, et al., Active cleaning of label noise, Pattern Recognition (2015), http://dx.doi.org/10.1016/j.patcog.2015.09.020