Robust Ensemble Classifier Combination Based on Noise Removal with One-Class SVM

Ferhat Özgür Çatak(✉)

TÜBİTAK BİLGEM, Cyber Security Institute, Kocaeli/Gebze, Turkey
ozgur.catak@tubitak.gov.tr

Abstract. In machine learning area, as the number of labeled input samples becomes very large, it is very difficult to build a classification model because of input data set is not fit in a memory in training phase of the algorithm, therefore, it is necessary to utilize data partitioning to handle overall data set. Bagging and boosting based data partitioning methods have been broadly used in data mining and pattern recognition area. Both of these methods have shown a great possibility for improving classification model performance. This study is concerned with the analysis of data set partitioning with noise removal and its impact on the performance of multiple classifier models. In this study, we propose noise filtering preprocessing at each data set partition to increment classifier model performance. We applied Gini impurity approach to find the best split percentage of noise filter ratio. The filtered sub data set is then used to train individual ensemble models.

Keywords: One-class SVM · Data partitioning · Noise filtering · Gini impurity · Large scale data classification

1 Introduction

It's clear that we collect and store larger amounts of data in databases. The need for efficiently and effectively analyzing and utilizing the information contained in the data has been increasing. Just as big data technologies evolved, the quantity and variety of data has also increased, and becoming more focused on storing every type of data. The main purpose of the storing of the data is intended to obtain information from data using a variety of machine learning methods. One of the primary machine learning techniques is classification, which labels the new samples based on a training set whose class labels are provided [1,2]. Classification methods are applied in various areas such as bioinformatics, pattern recognition, text mining, social network analysis, etc.

In Big Data age, traditional classifier algorithms have new challenges to scaling up in order to address the large-scale data set training. Most of existing classification algorithms assume that the data can fit in a memory in training phase of learning. These algorithms cannot be comfortably implemented to data sets that larger than computer memory capacity. Data partitioning strategy is

S. Arik et al. (Eds.): ICONIP 2015, Part II, LNCS 9490, pp. 10–17, 2015.
DOI: 10.1007/978-3-319-26535-3_2

one of the methods that can be applied to the training of high-dimensional data sets that are used for the building of classification model in order to overcome the input data complexity. In order to prevent the building of weak classification model that emerged from the data chunks, the input set needs to be strengthened through various methods.

In this study, the noise filtering approach is applied to each individual sub data set to clean noisy input data, then, AdaBoost ensemble method is used to strength the classification model at each data partition. We applied one-class Support Vector Machine (SVM) method to filter noisy instances from each individual data partition and then AdaBoost ensemble based classification method is used to each individual data partition to increase the model accuracy.

The overall contributions of the study are listed as follows:

1. Using data partitioning method, the complexity of input matrix, which is quite high for the single memory, is reduced in this manner.
2. Each individual sub-set of input matrix is reinforced with noise filtering method using one-class SVM and Gini impurity.
3. Each sub-set of input matrix is used in the training phase of the different ensemble classifier, so that each instances are considered when building a global classification model.

Gini impurity is used to calculate the uncertainty about source of input data set. This measure is applied to estimate the degree of information diversity provided by cleaned partition of sub data set.

The remainder of this paper is organized as follows: Sect. 2 briefly explains the methods that are used in this work. Section 3 describes the proposed data cleaning and partitioning method. Section 4 gives the experimental results. In Sect. 5, we give conclusion and future works.

2 Preliminaries

The approach presented in this paper uses one-class SVM algorithm to remove noisy instances, AdaBoost to build ensemble classifier models, and data partitioning to train over all data set instances. All elements are introduced here briefly.

2.1 One-Class SVM

SVM [3] method is used to find classification models using the maximum margin separating hyper plane. Schölkopf et al. [4] proposed a training methodology that handles only one class classification called as "one-class" classification.

One-class SVM algorithm is a method used to detect the outliers in the data. Basically the method finds soft boundaries of the data set, and then, model determines whether new instance belongs to this data set or not. Suppose, we are given a data set, $\mathbf{x}_1, \ldots \mathbf{x}_m \in X$ drawn from an unknown underlying probability distribution P. We are interested in estimating a set S such that the probability

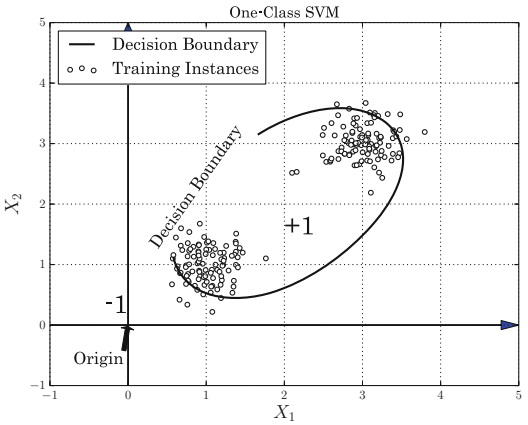


Fig. 1. One-class SVM. The origin, $(0, 0)$, is the single instance with label -1.

that a test point from P lies inside in S with an a priori specified probability value. As shown in Fig. 1, origin is labeled as -1, and the all training instances are labeled as $+1$.

Let $S = \{(\mathbf{x}_i, y_i)|\mathbf{x}_i \in \mathbb{R}^n, y \in \{1, \ldots, K\}\}_{i=1}^m$ be input instances in $\mathbb{R}^n$, $\phi : X \rightarrow H$ be a kernel function that maps the input instances to another space. Then standard SVM method tries to find a hyper plane that solves the separation problem with an optimization problem. The objective function of the SVM classifier is formulated as follows.

$$\begin{aligned} & min_{\mathbf{w},\xi,\rho} \left(\frac{1}{2}||\mathbf{w}||^2 + \frac{1}{mC} \sum_i \xi_i - \rho \right) \\ & subject\, to \\ & \qquad (w.\phi(\mathbf{x}_i)) \geq \rho - \xi_i \\ & \qquad \xi_i \geq 0, \forall i = 1, \ldots, m \end{aligned} \tag{1}$$

where $\mathbf{w}$ is orthogonal to the separating hyper plane, C is smoothness parameter, $\mathbf{x}_i$ is the i-th input instances, m is the total number of input instances, ξ_i are the slack variables, ρ is the distance between origin and separating hyper plane.

By using Lagrange techniques, $\mathbf{w}$ and ρ are obtained, then the decision function becomes:

$$f(\mathbf{x}) = sign\left((\mathbf{w}.\phi(\mathbf{x})) - \rho\right). \tag{2}$$

2.2 AdaBoost

The AdaBoost [5] is a supervised learning algorithm designed to solve classification problems [6]. The algorithm takes as input a training set $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)$ where the input sample $\mathbf{x}_i \in R^p$, and the output value, y_i, in a finite space

$y \in 1, ...K$. AdaBoost algorithm assumes a set of training data sampled independently and identically distributed (i.i.d.) from some unknown distribution $\mathcal{X}$.

Given a space of feature vectors X and two possible class labels, $y \in \{-1, +1\}$, AdaBoost goal is to learn a strong classifier $H(\mathbf{x})$ as a weighted ensemble of weak classifiers $h_t(\mathbf{x})$ predicting the label of any instance $\mathbf{x} \in X$ [7].

$$H(\mathbf{x}) = sign(f(\mathbf{x})) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right). \tag{3}$$

2.3 Data Partitioning Strategies

The use of multiple classifiers, learning methods are applied to base classifiers with different methods. Data partitioning is used a variety of reasons. First reason is the diversity that means uncorrelated base classifiers [8,9]. Another reason is the reducing the input complexity of large-scale data sets [10]. Last one is to build classifier models for the specific part of the input instances [11].

Data partitioning is basically divided into two different groups; filter based data partitioning and wrapper based data partitioning [12]. In wrapper based data partitioning, sub-data sets are created using base classifier outputs [13]. In filter based data partitioning, sub-data sets are created before individual classifiers are trained [14].

3 Proposed Approach

In this section we provide the details of the proposed noise filter based sub data set training method. The basic idea of noise removing based on one-class SVM technique is introduced in Sect. 3.1. The analysis of proposed method is described in Sect. 3.2.

3.1 Basic Idea

Our main task is to partition the input data set into sub-data sets, (X_m, Y_m), and, create local classifier ensembles for each sub data chunk. Noise removing process is applied to each individual sub-data set as pre-processing. Weighted voting method is used to combine the each ensemble classifier, and then, a single classifier model is created. Overall of the proposed method is shown in Fig. 2.

3.2 Analysis of the Proposed Algorithm

Kragh et al. showed that ensemble methods of neural networks gets better accuracy performance over unseen examples [15]. The main motivation of this work is the idea that small size classifier ensembles can obtain more accurate classifier model that are comparable to individual classifiers.

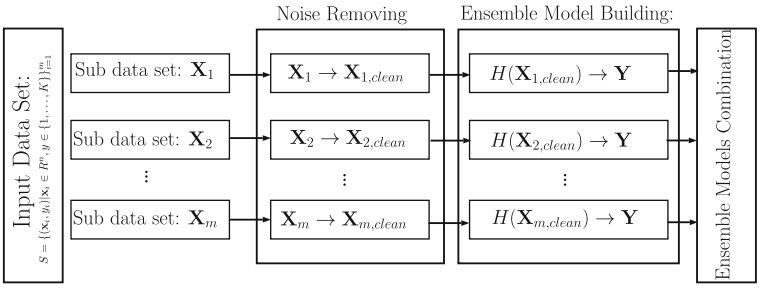


Fig. 2. Overall of proposed approach.

In the proposed model, at every sub-data set, there is a set of classifier functions (ensemble classifier), $H^{(m)}$, that acts as a single classification model. The single model at every sub-data set, m, is defined as follows:

$$H^{(m)}(\mathbf{x}) = \arg\max_{k} \sum_{t=1}^{M} \alpha_t h_t(\mathbf{x}) \tag{4}$$

The selected ensemble classifier models from last phase of our algorithm are combined into one single classification model, $\hat{H}(\mathbf{x})$, using accuracy based majority voting method.

$$\hat{H}(\mathbf{x}) = \arg\max_{k} \sum_{i=1}^{m} \beta H^{(m)}(\mathbf{x}) \tag{5}$$

where β is the accuracy of ensemble classifier.

4 Experiments

In this section, we perform experiments on real-world data sets from the public available data set repositories. Public data sets are used to evaluate the proposed learning method. Classification models of each data set are compared for accuracy results without removing noisy samples from them.

4.1 Experimental Setup

In this section, our approach is applied to five different data sets to verify its model effectivity and efficiency. The data sets are summarized in Table 1, including cod-rna, ijcnn1, letter, shuttle and SensIT Vehicle. We choose 50 as the data split size, m, and 3 different classification methods including Extra Trees [16], k-nn and SVM.

Table 1. Description of the testing data sets used in the experiments.

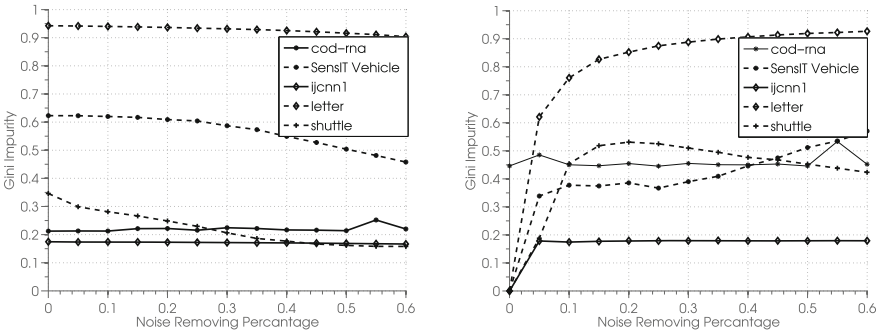
Data set	#Train	#Test	#Classes	#Attributes
cod-rna	59,535	157,413	2	8
ijcnn1	49,990	91,701	2	22
letter	15,000	5,000	26	16
shuttle	43,500	14,500	7	9
SensIT vehicle	78,823	19,705	3	100

4.2 Effect of Noise Removing on Input Matrix

In this section, we show the impact of noise removal pre-processing on the sample data sets. In order to show the noise removing affects, we used the "Gini Impurity" to measure the quality of procedure. Gini approaches deal appropriately with data diversity of a data. The Gini measures the class distribution of variable $\mathbf{y} = \{y_1, \cdots, y_m\}$. The Gini impurity can be written as:

$$g = 1 - \sum_{k} p_j^2 \tag{6}$$

where p_j is the probability of class k, in data set $\mathcal{D}$.



(a) Cleaned part. (b) Removed part.

Fig. 3. The impact of one-class SVM on the performance on selected data sets in terms of Gini impurity.

The cleaning results are shown in Fig. 3a and b. As expected, the Gini impurity value decreases with the cleaning of the noisy instances from the data, and increases on separated noisy data. Our aim is to minimizing the Gini impurity on clean data set, $\mathbf{X}_{clean}$, and maximizing the value on noisy data set $\mathbf{X}_{noisy}$. As a result, this division ratio which minimizes the ratio between the two values was regarded as the optimum value.

$$Split\,Percantage = \arg\max_{p} \frac{Gini(\mathbf{X}_{clean}, p)}{Gini(\mathbf{X}_{noisy}, p)} \tag{7}$$

Table 2 shows the best Gini impurity performances of each data set used in our experiments.

Table 2. The best noise removal percentages of each data sets.

Data sets	Percentage	$Gini(\mathbf{X}_{clean})$	$Gini(\mathbf{X}_{noisy})$	$\frac{Gini(\mathbf{X}_{clean})}{Gini(\mathbf{X}_{noisy})}$
cod-rna	0,55	0,331344656	0,56831775	0,583027112
SensIT vehicle	0,60	0,461155751	0,568847982	0,810683638
ijcnn1	0,60	0,167652825	0,179397223	0,934534117
letter	0,60	0,9039108	0,926430562	0,975691905
shuttle	0,30	0,214142833	0,506938229	0,422423918

4.3 Simulation Results

The process of the experiments are as follows: Firstly, we trained our data sets without using noise removal. Then we perform classification on test data sets, and calculate the accuracy of classifiers. We repeated the experiments 50 times, and average accuracy is calculated. Table 3 shows the average accuracy of each example data sets with and without noise removing using one-class SVM method.

As can be seen on Table 3, the noise removing based partitioned proposed algorithm significantly outperforms the splitted classifier building in most cases.

Table 3. Classification performance on example datasets using One-Class SVM noise removing and without removing for the proposed learning algorithm.

Data set	Extra trees		K-nn		SVM	
	All	Clean	All	Clean	All	Clean
cod-rna	0,75929	0,78652	0,91955	0,93513	0,88553	0,89806
ijcnn1	0,69758	0,72175	0,75533	0,77833	0,82989	0,84326
letter	0,91255	0,90853	0,90594	0,90318	0,92121	0,9199
shuttle	0,47801	0,44792	0,20262	0,26974	0,62301	0,63939
SensIT vehicle	0,9602	0,90558	0,87904	0,88266	0,99232	0,99174

5 Conclusions

In this paper, we have introduced a novel data partitioning based classifier building method, which improves the sub data sets with removing the noisy instances using one-class SVM and find best noise removing ratio with Gini impurity value. We carried out a series of computer experiments to find a global ensemble classifier and the performance of our proposed method. The training process of a partitioned data set is simple, fast and final classifier model handle overall training instances. Our experimental results show that the memory requirement of training phase reduced remarkably, and the accuracy increased by using the

noise removal process. The proposed method is a practical multiple ensemble classifier training model to classify large-scale data sets.

In the future work, our plan is to study different noise removing methods to clean sub data set. We plan adaptive noise removing ratio to make our method as autonomous as possible.

References

1. Anderson, J.R., Michalski, R.S., Carbonell, J.G., Mitchell, T.M.: Machine Learning: An Artificial Intelligence Approach, vol. 2. Morgan Kaufmann, San Mateo (1986)
2. Ramakrishnan, R., Gehrke, J.: Database Management Systems. Osborne/ McGraw-Hill, Berkeley (2000)
3. Vapnik, V.: The Nature of Statistical Learning Theory. Springer Science and Business Media, New York (2000)
4. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Comput. **13**(7), 1443–1471 (2001)
5. Freund, Y., Schapire, R.E.: A desicion-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P. (ed.) Computational Learning Theory. Lecture Notes in Computer Science, vol. 904, pp. 23–37. Springer, Heidelberg (1995)
6. Freund, Y., Schapire, R., Abe, N.: A short introduction to boosting. J.-Jpn. Soc. Artif. Intell. **14**(771–780), 1612 (1999)
7. Landesa-Vzquez, I., Alba-Castro, J.L.: Double-base asymmetric AdaBoost. Neurocomputing **118**, 101–114 (2013)
8. Kuncheva, L.I.: Using diversity measures for generating error-correcting output codes in classifier ensembles. Pattern Recogn. Lett. **26**(1), 83–90 (2005)
9. Dara, R.A., Makrehchi, M., Kamel, M.S.: Filter-based data partitioning for training multiple classifier systems. IEEE Trans. Knowl. Data Eng. **22**(4), 508–522 (2010)
10. Chawla, N.V., Moore, T.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P., Springer, C.: Distributed learning with bagging-like performance. Pattern Recogn. Lett. **24**(1), 455–471 (2003)
11. Woods, K., Bowyer, K., Kegelmeyer Jr., W.P.: Combination of multiple classifiers using local accuracy estimates. In: 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings CVPR 1996, pp. 391–396. IEEE (1996)
12. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. Artif. Intell. **97**(1), 245–271 (1997)
13. Freund, Y., Schapire, R.E., et al.: Experiments with a new boosting algorithm. ICML **96**, 148–156 (1996)
14. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
15. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. In: Advances in Neural Information Processing Systems, pp. 231–238. MIT Press (1995)
16. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **63**(1), 3–42 (2006)