Refining Classifier from Unsampled Data

Donghai Guan, Yong-Koo Han, Young-Koo Lee, Sungyoung Lee, and Chongkug Park

Abstract— **For a learning task with a huge number of training instances, we sample some informative/important instances, which are then used for learning. Obtaining accurately labeling data is always difficult thus noise detection is required to filter out noises from sampled instances since the noises will degrade the learning performance. In this work, we propose to utilize unsampled instances to improve the performance of noise detection in sampled instances. Empirical study validates our idea that refined classifier can be achieved from noisy sampled instances by utilizing unsampled instances.**

I. Introduction

INDUCTIVE learning algorithm aims to form a good generalization model based on training instances. Generally with the number of training instances increasing, better generalization model could be achieved. Therefore people would like to use all the available labeled instances for training purpose. However, for the learning task with a huge number of labeled instances, it is impossible to use all the instances for training considering the training time and storage. For example, for intrusion detection task, most researchers employ the KDD'99 intrusion detection dataset (743M) and always sample partial of them as training data [1, 2].

When we obtain the sampled training data, generally there are two main factors which determine the quality of generalization model: the quality of training data and the appropriateness of the biases of the chosen learning algorithm for the training data. When the learning algorithm is given, the quality of generalization model mainly depends on the quality of training data. Considering that training data usually include noises which tend to degrade the quality of generalization model, effective noise detection is one of the most important problems in inductive learning.

In order to minimize the downside of noisy training instances, people mainly take either of the two approaches: noise tolerance and noise detection. Noise tolerance tries to control the negative effect of noisy instances without removing them, usually by designing robust algorithms that are insensitive to noise. The typical methods in this category include rule truncation [3] and tree pruning [4]. On the contrary, noise detection tries to improve the quality of training data by identifying and eliminating the noisy instances prior to apply the learning algorithm. One typical method in the category is to use an ensemble of classifiers and treat the training instance that is misclassified as noise. It has been argued by [5] that the noise tolerance is less effective than noise detection. In this work, we focus on noise detection of sampled training data.

The noisy training instances mainly include two types: attribute noise and class noise. Attribute noises are the errors in the attribute values of the instances. For class noise, they are also called mislabeled noises since they are caused by the mislabeling. Quinlan [6] has comprehensively analyzed the two types of noises and demonstrated that, for higher levels of noise, removing noise from attribute information decreases the predictive accuracy of the resulting classifier if the same attribute noise present when the classifier is subsequently used. However, for class noise, the opposite is true: cleaning the training data will result in a classifier with a higher predictive accuracy. Our focus is identifying and eliminating class noises (mislabeled instances) from sampled instances, thereby increasing the classifier's predictive accuracy.

Up to now, many research efforts have been made on eliminating mislabeled instances for effective learning. Guyon [7] provided an approach that uses an information criterion to measure an instance's typicality; and atypical instances are then presented to a human expert to determine whether they are mislabeled instances or exceptions. The noise detection algorithm of Gamberger [8] is based on the observation that the elimination of noisy examples reduces the CLCH (Complexity of the Least Complex correct Hypothesis) value of the training set. They called their noise elimination algorithm the Saturation filter since it employs the CLCH measure to test whether the training set is saturated. Brodley and Friedl [9,10] simplified noise elimination as a filtering operation where multiple classifiers learned from noisy training data are used to identify noise, and the noise is characterized as the instances that are incorrectly classified by the multiple classifiers. Two major filtering methods they proposed are majority filtering and consensus filtering. In addition, there are some noise detection methods specially proposed for nearest neighbor classifiers. Wilson [11] used a three-nearest neighbor classifier (3-NN) to select instances that then used to form a 1-NN. Aha, Kibler, and Albert [12] demonstrated that filtering instances based on records of their contribution to classification accuracy in an instance-based classifier

Manuscript received June 1, 2009. This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)" (IITA-2009-(C1090-0902-0002)) and is supported by the Brain Korea 21 projects and was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (No. 2008-1342)

The authors are with the Computer Engineering Department, Kyung Hee University, Korea. Corresponding author is Professor Young-Koo Lee, email: yklee@khu.ac.kr

improves the accuracy of the resulting classifier.

To filter out noises from sampled instances, all the noise filtering methods deal with sampled instances directly and none of them consider utilizing unsampled instances. We argue that unsampled instances have potential utility to improve the performance of noise filtering in sampled instanced, and that unsampled instances should be considered in noise filtering although they will not be used as the final training instances. In this paper, we propose to utilize the unsampled instances to aid the noise detection in sampled instances.

To assess the benefit of our approach, two popular noise detection methods are chosen which are majority filtering (MF) and consensus filtering (CF). With the aid of unsampled instances, the variants of MF and CF are represented by MF+ and CF+. We evaluate their performances by a set of experiments. Empirical study shows that MF+ and CF+ can improve the performance of MF and CF under different noise levels from 10% to 40%.

The rest of the paper is organized as follows: Section 2 describes the related works on noise filtering including both the original methods and our proposed methods. Section 3 discusses the experiments by using benchmark data. Section 4 summarizes our conclusions and future work.

II. Related works on noise filtering

There are many noise filtering methods which can filter noises from sampled instances. Herein, we consider Brodley's majority filtering (MF) and consensus filtering (CF) due to their wide-spread and popular use in the literature.

The general idea of MF and CF is as follows: They employ ensemble classifier to detect mislabeled instances by constructing a set of base-level classifiers and then using their classifications to identify mislabeled instances. The reason to employ ensemble classifiers in MF and CF is that ensemble classifier has better performance than each base-level classifier on a dataset if two conditions hold: (1) the probability of a correct classification by each individual classifier is greater than 0.5 and (2) the errors in predictions of the base-level classifiers are independent. The general approach is to tag an instance as mislabeled if x of the m base-level classifiers cannot classify it correctly. MF tags an instance as mislabeled if more than half of the m base level classifiers classify it incorrectly. CF requires that all base-level classifiers must fail to classify an instance as the class given by its training label for it to be eliminated from the training data. Majority filtering and Consensus filtering algorithms are shown in Fig. 1 and Fig. 2.

With the knowledge of MF and CF, we now consider how to utilize unsampled instances to aid the noise filtering of sampled instances. In this work, our idea is straightforward: mixing sampled instances and unsampled instances together, which is then inputted to MF/CF. Finally the intersection of sampled instances and the filtered instances from MF/CF is used as training instances.

Algorithm 1: MajorityFiltering (MF)
Input: E (training set)
Parameter: n (number of subsets), y (number of learning algorithms)
$A_1, A_2, \ldots, A_y$ (y kinds of learning algorithms)
Output: A (detected noisy subset of E)
(1) form n disjoint almost equally sized subsets of E_i, where $\cup_i E_i = E$
(2) $A \leftarrow \emptyset$
(3) **for** i=1,…, n **do**
(4) form $E_t \leftarrow E \backslash E_i$
(5) **for** j=1,…,y **do**
(6) induce H_j based on examples in E_t and A_j
(7) **end for**
(8) **for** every $e \in E_i$ **do**
(9) $ErrorCounter \leftarrow 0$
(10) **for** j=1,…,y **do**
(11) **if** H_j incorrectly classifies e
(12) **then** $ErrorCounter \leftarrow ErrorCounter + 1$
(13) **end for**
(14) **if** $ErrorCounter > \frac{y}{2}$, **then** $A \leftarrow A \cup \{e\}$
(15) **end for**
(16) **end for**

Fig. 1. Majority filtering

Algorithm 2: ConsensusFiltering (CF)
Input: E (training set)
Parameter: n (number of subsets), y (number of learning algorithms)
$A_1, A_2, \ldots, A_y$ (y kinds of learning algorithms)
Output: A (detected noisy subset of E)
(1) form n disjoint almost equally sized subsets of E_i, where $\cup_i E_i = E$
(2) $A \leftarrow \emptyset$
(3) **for** i=1,…, n **do**
(4) form $E_t \leftarrow E \backslash E_i$
(5) **for** j=1,…,y **do**
(6) induce H_j based on examples in E_t and A_j
(7) **end for**
(8) **for** every $e \in E_i$ **do**
(9) $ErrorCounter \leftarrow 0$
(10) **for** j=1,…,y **do**
(11) **if** H_j incorrectly classifies e
(12) **then** $ErrorCounter \leftarrow ErrorCounter + 1$
(13) **end for**
(14) **if** $ErrorCounter = y$, **then** $A \leftarrow A \cup \{e\}$
(15) **end for**
(16) **end for**

Fig. 2. Consensus filtering

III. Empirical study

The main objective of the empirical study is to assess the benefit of unsampled instances for noise detection in sampled instances. Section 3.1 explains the experimental setup. Afterwards Section 3.2 presents the experimental results.

3.1 Experimental setup

Existing MF, CF, and our proposed MF+, CF+ are tested on the benchmark data sets from the Machine Learning Database Repository [13]. Information of these data sets is tabulated in Table 1. These data sets are collected from

different real-world applications in various domains. Each data set is divided into sampled set, unsampled set, and test set. Noise detection methods works on the sampled set and outputs the filtered set. Afterwards the test set is classified by the classifiers which are trained on the various filtered sets. Classification error rate is the measure to evaluate the performance of each noise detection method on the classifier, where

$$\text{classification error rate} = \frac{\text{No. of incorrect classifications on testing instances}}{\text{No. of testing instances}}$$

TABLE I
UCI DATA SETS USED IN THE EXPERIMENT

Data set	Attribute	Size	Class	Class distribution
Iris	4	150	3	50/50/50
Voting	16	435	2	267/168
heart2	13	294	2	188/106
Horse	15	368	2	232/136
Sonar	60	208	2	111/97
Wine	13	178	3	59/71/48
Breast	9	1000	2	700/300
australian	14	690	2	383/307
Bupa	6	345	2	145/200
Diabetes	8	768	2	500/268
Echo	7	131	2	88/43
German	24	1000	2	700/300
Glass	9	214	6	70/76/17/13/9/30
Credit	15	690	2	307/383
Spect	44	267	2	212/55
Wdbc	31	569	2	357/212
Ecoli	7	336	8	143/77/52/35/20/5/2/2
ionosphere	34	351	2	225/126

When two noise detection methods are applied to the same sampled set with the same learning algorithm, lower classification error rate indicates that the noise detection performance is better. To obtain the classification error rate, each data set D is processed as follows:

1) D is randomly partitioned into two parts: set T and set E_{us}. That is to say, random sampling is used as the sampling strategy.
2) Ten trials derived from ten-fold cross-validation on T are used to evaluate the performance of each noise detection method. At each trial, 90% of T is firstly selected and it is denoted by E representing the sampled instances. Most data sets here are experimental data sets where the ratio between noisy data to the whole data might be very small and even can be neglected. However, the performance of noise filtering method need to be evaluated on the noisy data sets. To this end, we artificially generate some noises in E by selecting some instances at random and then incorrectly changing their labels. The number of selected instances, that is the number of generated noises, is based on the defined noise ratio, which is the ratio between noisy data to the data in E.
3) E_{us} is used as the unsampled set. We artificially generate some noises in E_{us} by selecting some instances at random and then incorrectly changing their labels. The number of generated noises is based on the same noise ratio of sampled set E.
4) E is used as sampled set and it will be processed by MF, CF, MF+ (aided by E_{us}), and CF+ (aided by E_{us}) respectively. The remaining 10% of T is used as test set to evaluate the performance of various filtered sampled instances of E.
5) The average classification error rate is obtained by averaging ten trials' error rates.
6) Considering that the partition of sampled set and unsampled set could influence this average classification error rate, we execute the partition five times and get five classification error rates (execute step 1-5 five times).
7) Finally the reported error rate is the further averaged value of these five values.

In this experiment, the four noise detection methods on sampled instances follow the same configuration which is as follows: n, that is the number of subsets, is set to 5; y, that is the number of learning algorithms, is set to 3; A1, A2, and A3, representing three learning algorithms, refer to k-nearest neighbor (1-NN), naïve Bayes, and decision tree.

In addition to the parameters in the noise detection methods, there are two major parameters in above experiment flow which can influence the experiment. The first parameter determines data partitioning (step 1 of above experiment flow) and it is the ratio between set T to whole data, referred partitioning ratio. In the experiment, partitioning ratio is set to 0.5. The second and most important parameter determines the noise ratio of sampled set E (and unsampled set E_{us}) (step 2 of above experiment flow).

Considering that the labeling data obtained from the real applications might have different noise ratios, we have performed several experiments varying the noise ratio to make the experiments comprehensively. The explored noise ratios include 10%, 20%, 30%, and 40%.

3.2 Experimental results

In this section, the base learning algorithms that are used for evaluating the quality of sampled instances are 1-NN, naïve Bayes, and decision tree. In stead of presenting the results with details for all of them, we show the results of 1-NN for analyzing purpose. Then, we will summarize the results of naïve Bayes and decision tree.

In Table 2 we show the classification error for each data set of the classifiers formed by 1-NN tested using no filter, consensus filtering (CF), consensus filtering aided by unsampled instances (CF+), majority filtering (MF), and majority filtering aided by unsampled instances (MF+). The last row reports the average classification error across all the data sets of above classifiers. Table 2 shows for 1-NN, on average its performance on CF+ is better than CF (classification error: 0.215 vs. 0.222). Similarly its performance on MF+ is better than MF (classification error: 0.206 vs. 0.216).

TABLE II
EXPERIMENTAL RESULTS OF 1-NN WHEN NOISE RATIO IS 10%

Dataset	1-NN (noise ratio is 10%)				
	None	CF	CF+	MF	MF+
iris	0.161	0.066	0.057	0.076	0.057
vote	0.171	0.102	0.102	0.118	0.107
heart2	0.307	0.235	0.241	0.229	0.233
horse	0.309	0.265	0.261	0.230	0.230
sonar	0.269	0.274	0.232	0.285	0.264
wine	0.195	0.104	0.087	0.091	0.075
breast	0.157	0.066	0.071	0.062	0.050
australian	0.254	0.190	0.183	0.163	0.161
bupa	0.480	0.470	0.464	0.462	0.472
diabetes	0.331	0.279	0.289	0.273	0.278
echo	0.414	0.35	0.319	0.308	0.269
german	0.34	0.307	0.305	0.294	0.293
glass	0.371	0.355	0.341	0.407	0.371
credit	0.274	0.200	0.208	0.176	0.169
spect	0.310	0.330	0.310	0.307	0.3
wdbc	0.121	0.056	0.056	0.064	0.056
ecoli	0.294	0.197	0.181	0.184	0.165
ionos	0.179	0.151	0.158	0.161	0.163
Ave.	0.274	0.222	0.215	0.216	0.206

Table 3 shows the experimental results of 1-NN when noise ratio is 20%. We obtain the similar observations as before: MF+ defeats MF and CF+ defeats CF. In detail, the classification error rate comparison between CF+ and CF is 0.236 vs. 0.245. The classification error rate comparison between MF+ and MF is 0.218 vs. 0.227.

TABLE III
EXPERIMENTAL RESULTS OF 1-NN WHEN NOISE RATIO IS 20%

Dataset	1-NN (noise ratio is 20%)				
	None	CF	CF+	MF	MF+
iris	0.258	0.111	0.092	0.085	0.063
vote	0.263	0.152	0.142	0.128	0.125
heart2	0.333	0.245	0.216	0.210	0.208
horse	0.333	0.277	0.273	0.277	0.251
sonar	0.323	0.300	0.301	0.317	0.312
wine	0.185	0.064	0.047	0.030	0.035
breast	0.209	0.102	0.098	0.075	0.067
australian	0.304	0.211	0.214	0.177	0.176
bupa	0.444	0.443	0.417	0.474	0.415
diabetes	0.375	0.304	0.306	0.284	0.283
echo	0.386	0.361	0.339	0.283	0.3
german	0.372	0.334	0.335	0.325	0.306
glass	0.449	0.408	0.403	0.414	0.416
credit	0.303	0.221	0.224	0.195	0.181
spect	0.374	0.352	0.352	0.368	0.344
wdbc	0.221	0.103	0.097	0.075	0.066
ecoli	0.327	0.197	0.172	0.174	0.175
ionos	0.292	0.231	0.216	0.200	0.196
Ave.	0.319	0.245	0.236	0.227	0.218

Table 4 and 5 show the experimental results of 1-NN when noise ratios are 30% and 40%. These two tables further validate the superiority of CF+ and MF+.

TABLE IV
EXPERIMENTAL RESULTS OF 1-NN WHEN NOISE RATIO IS 30%

Dataset	1-NN (noise ratio is 30%)				
	None	CF	CF+	MF	MF+
iris	0.276	0.155	0.129	0.117	0.090
vote	0.288	0.209	0.192	0.168	0.147
heart2	0.371	0.275	0.240	0.243	0.214
horse	0.368	0.351	0.336	0.331	0.291
sonar	0.367	0.350	0.341	0.38	0.326
wine	0.293	0.162	0.150	0.098	0.108

breast	0.327	0.186	0.152	0.103	0.081
australian	0.365	0.273	0.283	0.226	0.229
bupa	0.431	0.443	0.429	0.463	0.474
diabetes	0.407	0.350	0.329	0.319	0.283
echo	0.411	0.347	0.364	0.311	0.331
german	0.416	0.357	0.36	0.316	0.328
glass	0.535	0.449	0.427	0.454	0.393
credit	0.392	0.300	0.320	0.249	0.255
spect	0.406	0.398	0.388	0.405	0.406
wdbc	0.314	0.178	0.165	0.116	0.091
ecoli	0.415	0.187	0.194	0.197	0.167
ionos	0.334	0.269	0.241	0.233	0.242
Ave.	0.373	0.291	0.280	0.263	0.248

TABLE IV
EXPERIMENTAL RESULTS OF 1-NN WHEN NOISE RATIO IS 40%

Dataset	1-NN (noise ratio is 40%)				
	None	CF	CF+	MF	MF+
iris	0.276	0.155	0.129	0.117	0.090
vote	0.288	0.209	0.192	0.168	0.147
heart2	0.371	0.275	0.240	0.243	0.214
horse	0.368	0.351	0.336	0.331	0.291
sonar	0.367	0.350	0.341	0.38	0.326
wine	0.293	0.162	0.150	0.098	0.108
breast	0.327	0.186	0.152	0.103	0.081
australian	0.365	0.273	0.283	0.226	0.229
bupa	0.431	0.443	0.429	0.463	0.474
diabetes	0.407	0.350	0.329	0.319	0.283
echo	0.411	0.347	0.364	0.311	0.331
german	0.416	0.357	0.36	0.316	0.328
glass	0.535	0.449	0.427	0.454	0.393
credit	0.392	0.300	0.320	0.249	0.255
spect	0.406	0.398	0.388	0.405	0.406
wdbc	0.314	0.178	0.165	0.116	0.091
ecoli	0.415	0.187	0.194	0.197	0.167
ionos	0.334	0.269	0.241	0.233	0.242
Ave.	0.373	0.291	0.280	0.263	0.248

Table 5 and 6 report the results for naïve Bayes and decision tree under different noise ratios. It shows the same experimental trend as 1-NN: by using unsampled instances, the performances of MF and CF are improved.

TABLE V
Experimental results of naïve Bayes under different noise ratios

Noise ratio	Naïve Bayes				
	None	CF	CF+	MF	MF+
10%	0.234	0.218	0.213	0.220	0.213
20%	0.248	0.240	0.235	0.233	0.230
30%	0.263	0.253	0.245	0.247	0.240
40%	0.309	0.303	0.293	0.297	0.277

TABLE VI
Experimental results of decision tree under different noise ratios

Noise ratio	Decision Tree				
	None	CF	CF+	MF	MF+
10%	0.246	0.236	0.236	0.231	0.229
20%	0.269	0.229	0.221	0.225	0.217
30%	0.329	0.277	0.261	0.262	0.240
40%	0.387	0.351	0.325	0.332	0.300

IV. CONCLUSIONS AND FUTURE WORKS

This article presents a new approach for identifying mislabeled instances from sampled instances, which considers the potential utility of unsampled instances and make use of them to improve the noise detection performance.

Majority filtering (MF) and consensus filtering (CF) are two popular noise detection methods and they are chosen to assess the benefit of our approach. The results of an empirical evaluation demonstrated that MF and CF could be improved with the aid of unsampled instances.

Our future work is to refine the usage of unsampled instances to make them contribute more to the noise filtering in sampled instances.

REFERENCES

[1] C. Shuyan and Z. Li, "Training Samples Selection Method in Intrusion Detection System," *Computer Science and Computational Technology, 2008. ISCSCT '08. International Symposium on*, 2008, pp. 610-612.

[2] C. Ray-I, L. Liang-bin, "Intrusion Detection by Backpropagation Neural Networks with Sample-Query and Attribute-Query," *International Journal of Computational Intelligence Research*, Vol. 3, 2007, pp. 6-10.

[3] J. Mingers, "An Empirical Comparison of Pruning Methods for Decision Tree Induction," *Machine Learning*, vol. 4, Nov. 1989, pp. 227-243.

[4] J.R. Quinlan, *C4.5: programs for machine learning*, Morgan Kaufmann Publishers Inc., 1993.
[5] D. Gamberger, N. Lavrac, S. Dzeroski, "Noise Detection and Elimination in Data Preprocessing: Experiments in Medical Domains," Applied Artificial Intelligence, vol. 14, 2000, pp. 205-223.
[6] J.R. Quinlan, "Introduction of decision trees," *Machine Learning*, vol. 1, 1986, pp. 81-106.
[7] I. Guyon, N. Matic, and V. Vapnik, "Discovering informative patterns and data cleaning," *Advances in knowledge discovery and data mining*, American Association for Artificial Intelligence, 1996, pp. 181-203.
[8] D. Gamberger, N. Lavrac, and C. Groselj, "Experiments with Noise Filtering in a Medical Domain," *Proceedings of the Sixteenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., 1999, pp. 143-151.
[9] C.E. Brodley and M.A. Friedl, "Identifying and eliminating mislabeled training instances," *In AAAI/IAAI*, vol. 1, 1996, pp. 799--805.
[10] C.E. Brodley, P. Uiversity, M.A. Friedl, B. Uiversity, and B.P. Edu, "Identifying Mislabeled Training Data," *Journal of Artificial Intelligence Research*, vol. 11, 1999, pp. 131--167.
[11] D.L. Wilson, "Asymptotic Properties of Nearest Neighbor Rules Using Edited Data," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 2, 1972, pp. 408-421.
[12] D.W. Aha, D. Kibler, and M.K. Albert, "Instance-Based Learning Algorithms," *Machine Learning*, vol. 6, Jan. 1991, pp. 37-66.
[13] A. Asuncion, D.J Newman, UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science.