Paper:

Data Cleaning for Classification Using Misclassification Analysis

Piyasak Jeatrakul, Kok Wai Wong, and Chun Che Fung

School of Information Technology, Murdoch University
South Street, Murdoch, Western Australia 6150, Australia
Email: {p.jeatrakul, k.wong, l.fung}@murdoch.edu.au
[Received December 9, 2009; accepted February 9, 2010]

In most classification problems, sometimes in order to achieve better results, data cleaning is used as a pre-processing technique. The purpose of data cleaning is to remove noise, inconsistent data and errors in the training data. This should enable the use of a better and representative data set to develop a reliable classification model. In most classification models, unclean data could sometime affect the classification accuracies of a model. In this paper, we investigate the use of misclassification analysis for data cleaning. In order to demonstrate our concept, we have used Artificial Neural Network (ANN) as the core computational intelligence technique. We use four benchmark data sets obtained from the University of California Irvine (UCI) machine learning repository to investigate the results from our proposed data cleaning technique. The experimental data sets used in our experiment are binary classification problems, which are German credit data, BUPA liver disorders, Johns Hopkins Ionosphere and Pima Indians Diabetes. The results show that the proposed cleaning technique could be a good alternative to provide some confidence when constructing a classification model.

Keywords: data cleaning, data pre-processing, artificial neural network, classifier

1. Introduction

In most classification or function approximation problems, the establishing of an accurate prediction model has always been a challenging problem. When constructing a prediction model, it is always difficult to have an exact function or separation that describes the relationship between the input vector, X and target vector, Y. However, a probabilistic relationship govern by joint probability law ν can be used to describe the relative frequency of occurrence of vector pair (X_n, Y_n) for n training set. The joint probability law ν can further separate into environmental probability law μ and conditional probability law γ. For notation expression, the probability law can be expressed as:

$$P(\nu) = P(\mu)P(\gamma) \quad \ldots\ldots\ldots\ldots \quad (1)$$

For environmental probability law μ, it describes the occurrence of X. As for conditional probability law γ, it describes the occurrence of Y given X. A vector pair (X,Y) is considered as noise if X does not follow the environmental probability law μ, or the Y given X does not follow the conditional probability law γ.

According to Zhu and Wu [1], the performance of classification depends on two significant factors: the quality of the training data and the competence of learning algorithm. Therefore, a possible approach to enhance the performance in any type of classification systems is by improving the quality of training data. Generally, noise can be divided into two major types: *attribute noise* and *class noise* [1]. Attribute noise is related to the errors in the attributes such as missing values and redundant data, while class noise is the class error of instances. In addition, there are two categories of class noise: inconsistent error and misclassification error. Inconsistent error occurs when two similar instances belonging to different (or conflicting) classes, and misclassification error is found when instances are classified into the wrong classes.

In this paper, we propose a new technique of noise detection and elimination. We only concentrate on the class noise or misclassification error here. The core techniques used in our study is based on Artificial Neural Networks (ANNs).

In recent years, there are several studies on noise detection and elimination for improving the quality of training instances on classification systems. For example, Brodley and Friedl [2] proposed their approaches to identify and eliminate misclassification errors from the training dataset. They evaluated and compared the classification accuracy using three noise filtering techniques: a single algorithm, majority voting and consensus voting. The results asserted that after removing the class noise from the training set, the classification accuracies improved significantly. Miranda et al. [3] compared three techniques for noise detection and elimination in bioinformatics data sets. The three techniques are removal of noise instances, reclassifying noise instances, and a hybrid of removal and reclassifying techniques. They concluded that the noise removal technique provided more accurate classification

than the other two techniques: reclassifying and the hybrid method. Verbaeten and Assche [4] applied ensemble methods to identify and remove noisy instances from the training set in classification tasks. These methods are cross-validated committees, bagging and boosting for preprocessing. They also used the consensus and the majority voting techniques to identify and clean up misclassifications from the training set. They found that majority voting filters and bagging majority voting filters provided good results. However, more data sets are needed to be tested with these techniques. Zhu et al. [5] proposed a new technique called *Partitioning Filter (PF)* to remove misclassifications from large datasets. The results showed that at any noise level, the training sets that were filtered by Partition Filter always presented significantly improved classification accuracy when compared to the outcomes by using unclean datasets. Furthermore, Libralon et al. [6] applied distance-based techniques mainly to detect and remove noisy instances from the training dataset. Mislabeled tissues were detected and removed in gene expression classification problems. The results of the experiments showed that the performance of the classifiers were better when compared to the classification results by using the original datasets. Moreover, Tomek Links algorithm [7] which is a form of the *k*-Nearest Neighbor (*k*-NN) algorithm was applied as a data cleaning method in order to remove the noisy and borderline instances from the training set. Tomek links were identified by a 1-NN classifier if a pair of instance is belonging to different classes. This data cleaning technique has been used in several experiments. For example, Sun et al. [8] applied Tomek links technique to remove noisy data for improving binding site predictions on sequences of DNA. They concluded that by removing Tomek links from the training data, the classifier can improve the classification accuracy especially on the imbalanced data set.

Most of reported research try to increase the quality of training data by using some form of pre-processing. They are focusing on examining feasibility of effective techniques to reduce noise and enhance the performance of classification systems. This is thus the direction of this paper to move one step forward in misclassification analysis to improve the classification accuracy.

In this paper, we formulate a technique to perform misclassification analysis with an intention that we can identify noisy data with some confidence. After identifying the noisy data, we can then perform data cleaning. We apply the concept from the Complementary Neural Network (CMTNN) [9] as the cleaning technique to enhance the performance of a neural network classifier. CMTNN is selected because of its particular characteristics. It can integrate the truth and false membership values to deal with the uncertainty in classification while other techniques use only truth membership values.

In the experiments, four binary classification data sets from the University of California Irvine (UCI) machine learning repository [10] are used. These include German credit data, BUPA liver disorders, Johns Hopkins Ionosphere and Pima Indians Diabetes. These data sets are selected because they are benchmark data sets which have been commonly used in the literature. Finally, we compare the results of the proposed technique to the Tomek links cleaning, majority voting and consensus voting filtering techniques. In addition, these techniques are selected for this comparison because they has been applied to remove noisy effectively in several experiments [2, 4, 11, 12].

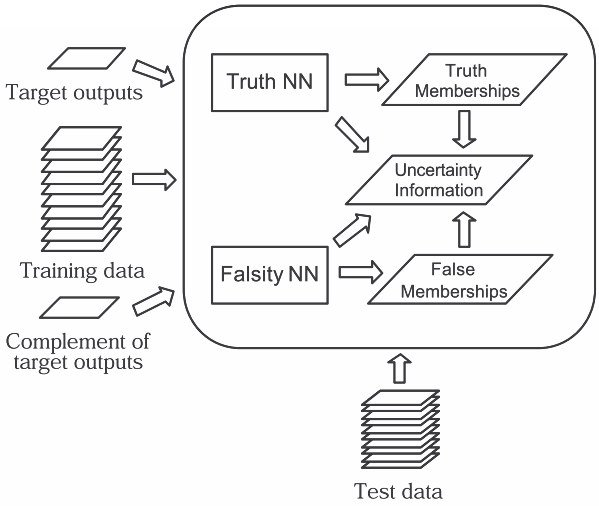

Fig. 1. Complementary neural network [13].

2. Cleaning Techniques Using Misclassification Analysis

In this section, the concept of Complementary Neural Network (CMTNN) is described and the proposed cleaning techniques based on CMTNN will then be presented.

2.1. Complementary Neural Network (CMTNN)

CMTNN [9] is a technique using a pair of complementary feedforward backpropagation neural networks called Truth Neural Network (Truth NN) and Falsity Neural Network (Falsity NN) as shown in **Fig. 1**.

While the Truth NN is a neural network that is trained to predict the degree of the truth memberships, the Falsity NN is trained to predict the degree of false memberships. Although the architecture and input of Falsity NN are the same as the Truth NN, Falsity NN uses the complement of target outputs of the Truth NN to train the network. For example, in binary classification problem, if the target output used to train the truth neural network is 0, the complement of this target output to train the falsity neural network will be 1. In the testing phase, the test set is applied to both networks to predict the degree of truth and false membership values. For each input pattern, the prediction of false membership value is expected to be the complement of the truth membership value [13].

Instead of using only the truth membership to classify the data, which is normally done by most convention neural network, the predicted results of Truth NN and Fal-

sity NN are compared in order to provide the classification outcomes. The difference between the truth and false membership values can also be used to represent uncertainty in the classification [14].

2.2. The Proposed Cleaning Techniques

In order to apply CMTNN for data cleaning, Truth NN and Falsity NN are employed to detect and clean misclassification patterns from a training set. The steps of our cleaning technique are described as follows.

1. The Truth NN and Falsity NN are trained by truth and false membership values.

2. The prediction outputs (Y) on the training data $(T)$of both NNs are compared with the actual outputs (O). The misclassification patterns of Truth NN and Falsity NN $(M_{\text{Truth}}, M_{\text{Falsity}})$ are also detected if the prediction outputs and actual outputs are different.

   For Truth NN : If $Y_{\text{Truth } i} \neq O_{\text{Truth } i}$ then

   $$M_{\text{Truth}} \leftarrow M_{\text{Truth}} \quad U \quad \{T_i\} \quad . \quad . \quad . \quad . \quad . \quad . \quad (2)$$

   For Falsity NN : If $Y_{\text{Falsity } i} \neq O_{\text{Falsity } i}$ then

   $$M_{\text{Falsity}} \leftarrow M_{\text{Falsity}} \quad U \quad \{T_i\} \quad . \quad . \quad . \quad . \quad . \quad (3)$$

3. In the last step, the new training set(T_c) is cleaned by eliminating the misclassification patterns detected by both the Truth NN (M_{Truth}) and Falsity NN (M_{Falsity}).

   $$T_c \leftarrow T - (M_{\text{Truth}} \cap M_{\text{Falsity}}) \quad . \quad . \quad . \quad . \quad . \quad . \quad (4)$$

As for training a new neural network classifier, the cleaned data set that removes those misclassification patterns will be used.

3. Experiments and Results

Four data sets from UCI machine learning repository [10] are used in the experiment. The data sets for binary classification problems include German credit data, BUPA liver disorders, Johns Hopkins Ionosphere and Pima Indians Diabetes.

- The purpose of German credit data set is to predict whether a loan application is “Good” or “Bad” credit risk.
- The purpose of BUPA liver disorders data set is to predict whether a male patient shows signs of liver disorders.
- The purpose of Johns Hopkins Ionosphere data set is to predict “Good” or “Bad” radar return from the ionosphere.
- The purpose of Pima Indians Diabetes data set is to predict whether a patient shows signs of diabetes.

Table 1. Characteristics of data sets used in the experiment.

Name of data set	No. of patterns	No. of attributes	No. of patterns in class 1	No. of patterns in class 2
German credit data	1000	20	700	300
BUPA liver disorders	345	6	200	145
Johns Hopkins Ionosphere	351	34	225	126
Pima Indians Diabetes	768	8	500	268

Table 2. Number of patterns in the training and test sets.

Name of data set	No. of training data	No. of test data	Total
German credit data	800	200	1000
BUPA liver disorders	276	69	345
Johns Hopkins Ionosphere	281	70	351
Pima Indians Diabetes	614	154	768

The characteristics of these three data sets are shown in **Table 1**.

For the purpose of establishing the classification model and testing it, each data set is first split into 80% training set and 20% test set as shown in **Table 2**. Furthermore, the cross validation method is used to obtain reasonable results. Each data set will be randomly split ten times to form different training and test data sets. For the purpose of this study, the results of the ten experiments of each data set will be averaged.

For our proposed cleaning technique, we create Truth NN and Falsity NN to detect the class noise using MATLAB version 7.4. These experimental conditions are shown in **Table 3**.

Table 4 shows the average number of misclassification patterns in each data set detected by Truth NN and Falsity NN. The results show that the number of misclassification patterns detected by both NNs is almost similar. For example, in German credit data, misclassification patterns detected by Truth NN and Falsity NN are 169 and 165 patterns respectively. Furthermore, there are also misclassification patterns discovered by both NNs, i.e., the same patterns that are misclassified by Truth NN as well as the Falsity NN. They are 125, 55, 6, 155 such patterns for German credit, BUPA liver disorders, John Hopkins Ionosphere and Pima Indians Diabetes data set respectively.

After the training sets are cleaned by the proposed cleaning technique as mentioned in section 2, new neural network classifiers are trained by the cleaned training sets. The performance of each classifier for the training set and test set before and after cleaning data are evalu-

Table 3. Configuration of neural networks in the experiments.

Experimental conditions	Values
Number of hidden layers	1
Learning rate	0.01
Number of hidden neurons	2 x number of input attributes
Transfer function	Log-sigmoid
Maximum time to train	Infinity
Momentum constant	0.7
Minimum performance gradient	1e-10
Performance goal	1e-3
Maximum validation failures	5
Maximum number of epochs to train	5000

Table 4. Average number of misclassification patterns of the training sets.

Name of data set	No. of misclassification patterns detected by Truth NN	No. of misclassification patterns detected by Falsity NN	No. of the misclassification patterns detected by both NNs
German credit data	169	165	125
BUPA liver disorders	79	77	55
Johns Hopkins Ionosphere	10	7	6
Pima Indians Diabetes	131	130	115

Table 5. Average classification accuracy (%) of the test sets before and after cleaning data.

Name of data set	Before cleaning	After cleaning training data with Tomek links technique	After cleaning training data with the majority voting filtering	After cleaning training data with the consensus voting filtering	After cleaning training data with the proposed technique
German credit data	76.25	77.45	76.35	77.00	77.55
BUPA liver disorders	69.99	70.72	70.29	71.01	71.45
Johns Hopkins Ionosphere	90.29	88.85	88.56	92.00	92.00
Pima Indians Diabetes	76.17	74.85	76.23	76.36	76.62

ated. Furthermore, in order to evaluate the performance of our proposed technique, we also compare the results of our proposed technique to other cleaning algorithms which are Tomek links cleaning, majority voting and consensus voting filtering techniques.

For the Tomek links cleaning technique, in order to identify a pair of instance which is belonging to different classes, the nearest neighbour method is used to find a Tomek link pair. Then, the noisy and borderline instances are cleaned from the training set.

For the majority voting and consensus voting filtering, we compare misclassification patterns detected by three different classification algorithms including ANN, Decision Tree (DT) and *k*-Nearest Neighbor (*k*-NN). In addition, DT and *k*-NN classifiers are created by SPSS Statistics Version 17.0. In the experiment, we applied a heuristic method for *k*-NN, the value of *k* used in *k*-NN classifier is considered as five. In order to apply the majority voting filtering, an instance is removed when it is misclassified by two out of three classifiers. Furthermore, if an instance is misclassified by all three classification algorithms, it is considered as noise for consensus voting.

The comparison results before and after cleaning by each technique for four data sets are shown in **Table 5**. It shows that our proposed cleaning technique outperforms other cleaning techniques in all cases. It performs best on all test sets while the consensus voting filtering performs second best.

The classification accuracies using our cleaning technique increases from 76.25% to 77.55% on German credit data, from 69.99 to 71.45% on BUPA liver disorders data, from 90.29% to 92% on Johns Hopkins Ionosphere, and from 76.17% to 76.62% on Pima Indians Diabetes. Furthermore, not every cleaning technique can perform well on any test sets. While Tomek links technique can only improve the classification performance on two datasets: German credit data and BUPA liver disorders data, the majority voting technique performs well on three out of four datasets.

In **Table 6**, the percents of misclassification patterns removed from the training set by each cleaning technique are compared in order to explain why our technique outperforms other cleaning techniques. It can be observed that the average percentage of misclassification patterns removed by Tomek links technique is the highest (27.89%). It is almost double when comparing to the patterns removed by our proposed technique (14.10%). Furthermore, our proposed technique removes misclassification patterns in the average percentage between the majority voting (21.81%) and consensus voting technique (9.10%).

From the observation, it can suggest that our technique removes only the highly possible misclassification patterns rather than eliminating all possible misclassification patterns as Tomek links and the majority voting techniques have performed, or removes only the most confi-

Table 6. Average misclassification patterns (%) removed from the training sets.

Name of data set	Tomek links	Majority voting filtering	The proposed technique	Consensus voting filtering
German credit data	29.58	22.41	15.59	11.7
BUPA liver disorders	38.19	31.41	20.07	9.63
Johns Hopkins Ionosphere	13.97	10.96	2.06	2.63
Pima Indians Diabetes	29.82	22.46	18.67	12.46
Average	27.89	21.81	14.10	9.10

dent patterns as the consensus voting technique has done. In other words, the Tomek links and the majority voting technique have the probability to clean out good patterns while the consensus voting technique is too conservative and it has a high probability to retain bad patterns.

In some cases, when the percentages of misclassification patterns of the two techniques are almost similar, the classification accuracies obtained by those techniques are the same as well. For example, in the experiment on Johns Hopkins Ionosphere data, our cleaning technique and consensus voting technique remove misclassification patterns by almost the same amount, 2.06% and 2.63% respectively. The classification accuracies obtained by both techniques are 92%. It asserts that the amount of noise patterns cleaned is a major factor affecting the quality of training data.

Although the improvement of the accuracies in this case study may not be significant, the proposed technique is able to provide a mean to increase the confidence of identifying the noisy data when compare to other cleaning techniques. It is still worth cleaning the noisy training data before it is learned by the classifier. There are also many factors that can be optimized in future to study the behaviour of the proposed misclassification analysis. The proportion of separating the training and testing data may be re-distributed to investigate the distribution of the training and testing set. Another danger for cleaning the noisy data is overtraining the ANN, more rigid generalization techniques could be experiment to study the behaviour of the model after the noisy data have been removed.

4. Conclusions

This paper presents the proposed misclassification technique to increase the confidence of cleaning noisy data used for training. In this paper, we focus our study for classification problem using ANN. The CMTNN is applied to detect misclassification patterns. For our proposed technique, the training data is cleaned by eliminating the misclassification patterns discovered by both the Truth NN and Falsity NN. After misclassification patterns are removed from the training set, a neural network classifier is trained by using the cleaned data. In the experiment of this paper, four data sets from the University of California Irvine (UCI) machine learning repository including German credit data, BUPA liver disorders, Johns Hopkins Ionosphere, and Pima Indians Diabetes are used. The neural network classifiers are evaluated and compared in terms of their performances. Furthermore, the results of the proposed technique are compared with other techniques including the Tomek links cleaning, the majority voting and the consensus voting filtering techniques. Results obtained from the experiment indicated this study could be carried further to optimize the misclassification analysis to be used as an alternative to improve the classification model.

References:

[1] X. Zhu and X. Wu, "Class Noise vs. Attribute Noise: A Quantitative Study," Artificial Intelligence Review, Vol.22, pp. 177-210, 2004.

[2] C. E. Brodley and M. A. Friedl, "Identifying mislabeled training data," J. of Artificial Intelligence Research, Vol.11, pp. 137-167, 1999.

[3] A. Miranda, L. Garcia, A. Carvalho, and A. Lorena, "Use of Classification Algorithms in Noise Detection and Elimination," in Hybrid Artificial Intelligence Systems, pp. 417-424, 2009.

[4] S. Verbaeten and A. Van Assche, "Ensemble methods for noise elimination in classification problems," in Multiple Classifier Systems, pp. 317-325, 2003.

[5] X. Zhu, X. Wu, and Q. Chen, "Eliminating Class Noise in Large Datasets," in Proceedings of the Twentieth Int. Conf. on Machine Learning (20th ICML), Washington D.C., pp. 920-927, 2003.

[6] G. L. Libralon, A. C. P. d. L. F. d. Carvalho, and A. C. Lorena, "Pre-Processing for Noise Detection in Gene Expression Classification Data," J. of Brazilian Computer Society, Vol.15, pp. 3-11, 2009.

[7] I. Tomek, "Two Modifications of CNN," Systems, Man and Cybernetics, IEEE Trans. on, Vol.6, pp. 769-772, 1976.

[8] Y. Sun, M. Robinson, R. Adams, R. T. Boekhorst, A. G. Rust, and N. Davey, "Using Sampling Methods to Improve Binding Site Predictions," in European Symposium on Artificial Neural Networks (ESANN'2006), Bruges, Belgium, 2006.

[9] P. Kraipeerapun, C. C. Fung, and S. Nakkrasae, "Porosity prediction Using Bagging of Complementary Neural Networks," in Advances in Neural Networks – ISNN 2009, pp. 175-184, 2009.

[10] A. Asuncion and D. J. Newman, "UCI Machine Learning Repository," University of California, Irvine, School of Information and Computer Sciences, 2007.

[11] T. W. Liao, "Classification of Weld Flaws with Imbalanced Class Data," Expert Systems with Applications, Vol.35, pp. 1041-1052, 2008.

[12] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data," SIGKDD Explor. Newsl., Vol.6, pp. 20-29, 2004.

[13] P. Kraipeerapun and C. C. Fung, "Binary Classification Using Ensemble Neural Networks and Interval Neutrosophic Sets," Neurocomput., Vol.72, pp. 2845-2856, 2009.

[14] P. Kraipeerapun and C. C. Fung, "Comparing Performance of Interval Neutrosophic Sets and Neural Networks with Support Vector Machines for Binary Classification Problems," in Digital Ecosystems and Technologies, 2008 (DEST 2008), 2nd IEEE Int. Conf. on, pp. 34-37, 2008.

Name:
Piyasak Jeatrakul

Affiliation:
Ph.D. Candidate, School of Information Technology, Murdoch University

Address:
South Street, Murdoch, Western Australia 6150, Australia

Brief Biographical History:
1996 Received B.Eng. Degree from King Mongkut's Institute of Technology Ladkrabang, Thailand
1999 Received M.B.A. Degree from National Institute of Development Administration, Thailand
2009 Received Postgraduate Diploma in Information Technology, Murdoch University, Australia

Main Works:
• P. Jeatrakul and K. W. Wong, "Enhance the Performance of Complementary Neural Network Using Misclassification Analysis," in Proc. of the Tenth Postgraduate Electrical Engineering and Computing Symposium (PEECS 2009), Perth, Australia, October 1, 2009.

Membership in Academic Societies:
• Institute of Electrical and Electronics Engineers (IEEE)

Name:
Kok Wai Wong

Affiliation:
Associate Professor, School of Information Technology, Murdoch University

Address:
South St, Murdoch, Western Australia 6150, Australia

Brief Biographical History:
2003- Nanyang Technological University
2005- Murdoch University

Main Works:
• Y. S. Ong, M. H. Lim, N. Zhu, and K. W. Wong, "Classification of Adaptive Memetic Algorithms: A Comparative Study," IEEE Trans. of Systems, Man, and Cybernetics, Part B: Cybernetics, Vol.36, No.1, February 2006, pp. 141-152, 2006.
• Z. C. Johanyak, D. Tikk, S. Kovacs, and K. W. Wong, "Fuzzy Rule Interpolation Matlab Toolbox - FRI Toolbox," Proc. of IEEE Int. Conf. on Fuzzy Systems 2006, July 2006, Vancouver, Canada, pp. 1427-1433, 2006.
• K. W. Wong, D. Tikk, T. D. Gedeon, and L. T. Koczy, "Fuzzy Rule Interpolation for Multidimensional Input Spaces with Applications," IEEE Trans. of Fuzzy Systems, Vol.13, No.6, December 2005, pp. 809-819, 2005.

Membership in Academic Societies:
• IEEE, IEEE Computational Intelligence Society, IEEE Computer Society
• Association for Computing Machinery (ACM)
• Australian Computer Society (ACS)

Name:
Chun Che Fung

Affiliation:
Associate Professor, School of Information Technology, Murdoch University

Address:
South Street, Murdoch, Western Australia, Australia 6150, Australia

Brief Biographical History:
1982-1988 Department of Electronic and Communication Engineering, Singapore Polytechnic
1989-2002 School of Electrical and Computer Engineering, Curtin University
2003- Joined Murdoch University

Main Works:
• "Simulated-Annealing-Based Economic Dispatch Algorithm," IEE Proc., Part C, Generation, transmission and distribution, Vol.140, No.6, pp. 509-515, 1993.
• "Modular Artificial Neural Network for Prediction of Petrophysical Properties from Well log data," IEEE Trans. on Instrumentation and Measurement, Vol.46, No.4, pp. 1295-1300, 1997.
• "The STAG Oilfied Formation Evaluation: a Neural Network Approach," Australian Petroleum Production & Exploration J., Vol.39, part 1, pp. 451-460, 1999.

Membership in Academic Societies:
• Senior Member, Institute of Electrical and Electronics Engineers (IEEE)
• Member, Institute of Engineer Australia (IEAust)
• Member, Australian Computer Society (ACS)