Pruning Training Samples Using a Supervised Clustering Algorithm

Minzhang Huang[1], Hai Zhao[1,2], and Bao-Liang Lu[1,2,★]

[1] Center for Brain-Like Computing and Machine Intelligence
Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems
Shanghai Jiao Tong University
800 Dong Chuan Rd., Shanghai 200240, China
{zhaohai,blu}@cs.sjtu.edu.cn

Abstract. As practical pattern classification tasks are often very-large scale and serious imbalance such as patent classification, using traditional pattern classification techniques in a plain way to deal with these tasks has shown inefficient and ineffective. In this paper, a supervised clustering algorithm based on min-max modular network with Gaussian-zero-crossing function is adopted to prune training samples in order to reduce training time and improve generalization accuracy. The effectiveness of the proposed training sample pruning method is verified on a group of real patent classification tasks by using support vector machines and nearest neighbor algorithm.

Keywords: Supervised clustering, Min-max modular network, Gaussian-zero-crossing function, Patent classification, Training sample pruning.

1 Introduction

More than one million new patent applications are submitted every year. It is a key problem to automatically classify these incoming patent applications. Currently most patents are handled in a manual way. For a very large-scale patent database, automatic classification approach may play an important role of effectively reducing the workload. Naive Bayes, *k*-NN, support vector machines (SVMs) and decision tree have been successfully applied to patent classification, and SVMs have been shown the best performance [3].

Patent classification task is often not only very large-scale but also serious imbalance. As a result, applying traditional pattern classification techniques becomes unacceptable for both training time and space complexities or the classification accuracy. Our solution in this work is to reduce those redundant or unreliable training samples by introducing a supervised clustering algorithm based on min-max modular network with Gaussian-zero-crossing function.

Clustering is a method for dividing data into several non-overlapped parts, i.e., clusters. A basic hypothesis about clusters is that data in the same cluster are more similar

[★] Corresponding author.

L. Zhang, J. Kwok, and B.-L. Lu (Eds.): ISNN 2010, Part II, LNCS 6064, pp. 250–257, 2010.

than data in different clusters. There are three types of clustering methods, supervised, semi-supervised and unsupervised clustering. Unsupervised clustering is a learning framework using a specific object functions, for example, a function that minimizes the distances inside a cluster to keep the cluster tight. Supervised clustering is applied on classified examples with the objective of identifying clusters that have high probability density with respect to a single class. Semi-supervised clustering is to enhance a clustering algorithm by using side information in clustering process. It can be subdivided into two major groups: similarity-based methods and search-based methods [2]. Fig. 1 illustrates the differences among these three types of clustering methods. In Fig. 1 (b), all the given data are unlabeled. In Fig. 1 (c), only the data have a outer circle are labeled. In Fig. 1 (d), all the data are labeled.

For supervised clustering, the class information of each data is available, most commonly used methods are learning vector quantization (LVQ) [5] and correlation clustering [1]. The simplest clustering action is to add the data into an existing cluster which closest to it or let the data to be a new cluster itself. But it is still a problem of how to discriminate these two circumstances. Li and Ye have proposed a supervised clustering method to solve this problem [7,8]. They divide the input space into several grids, only those data in the same grid can be grouped into the same cluster. But how to define the size of the grid is still unsolved. In our previous work, we have proposed a new supervised clustering method to overcome this difficulty by using min-max modular network with Gaussian-zero-crossing function [6].

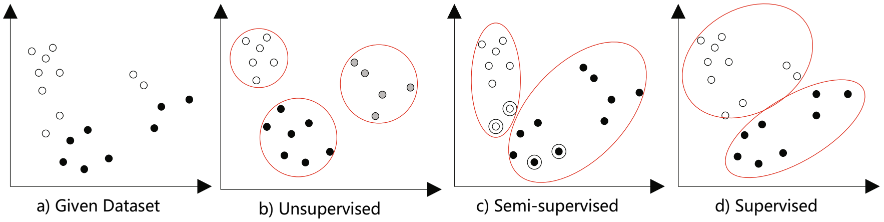

Fig. 1. Three different clustering methods: (a) given dataset; (b) unsupervised clustering; (c) semi-supervised clustering; and (d) supervised clustering

The min-max modular network with Gaussian-zero-crossing function (M^3-GZC) [9] is a special case of the min-max modular network (M^3-network) [10]. GZC function is directly adopted to distinguish two samples from different categories. In this paper, a supervised clustering algorithm based on M^3-GZC is introduced to prune training samples.

The reminder of this paper is organized as follows. In Section 2, M^3-network and M^3-GZC network are briefly introduced and the supervised clustering algorithm based on M^3-GZC network is described. The experiment results is presented in Section 3. Finally, the last section concludes this paper.

2 Supervised Clustering Algorithm

2.1 Min-Max Modular Network

The process of constructing an M^3-network consists of three steps: a) divide a complex K-class problem into several small independent two-class problems; b) solve these small problems in parallel, and c) finally integrate them according to two principles, namely the minimization principle and maximization principle [10].

Let T be the training set for a K-class classification problem:

$$T = \{X_l, D_l\}_{l=1}^{L},$$

where $X_l \in R^n$ is the input feature vector, D_l is the expected output vector, and L is the number of training samples.

The original K-class problem can be divided into $K(K-1)/2$ two-class subproblems as follows:

$$T_{ij} = \{(X_l^{(i)}, 1-e)\}_{l=1}^{L_i} \cup \{(X_l^{(j)}, e)\}_{l=1}^{L_j} \text{ for } i = 1, 2, ..., K \text{ and } j = i+1, ..., K \tag{1}$$

All of the the two-class problems defined in Eq. (1) can be trained in parallel. Their outputs are combined by the minimization principle:

$$M_i(x) = \min_{j=1}^{K} M_{ij}(x) \tag{2}$$

where $M_{ij}(x)$ represents the discriminative function of the component classifier trained on T_{ij}. If a two-class problem defined in Eq. (1) is still large and imbalance, it can be further divided. Assume that X_i in (1) is divided into N_i subset:

$$X_{ij} = \{X_l^{(ij)}\}_{l=1}^{L_i^{(j)}} \text{ for } j = 1, 2, ..., N_i \tag{3}$$

where $L_i^{(j)}$ denotes the number of training samples belonging to subset X_{ij}, and $\cup_{j=1}^{N_i} X_{ij} = X_i$.

According to Eq. (3, each of the two-class problem defined in Eq. (1) can be decomposed into the following relatively smaller two-class subproblems:

$$\begin{gathered} T_{ij}^{(u,v)} = \{(X_l^{(iu)}, 1-e)\}_{l=1}^{L_i^{(u)}} \cup \{(X_l^{(jv)}, e)\}_{l=1}^{L_j^{(v)}} \\ \text{for } u = 1, ..., N_i, \ v = 1, ..., N_j, \ i = 1, ..., K \text{ and } j \neq i \end{gathered} \tag{4}$$

where $X_l^{(iu)} \in X_{iu}$ and $X_l^{(jv)} \in X_{jv}$ are the input vectors belonging to C_i and C_j, respectively.

The solution to the original problem can be obtained by combining all of the trained component classifiers as follows:

$$\begin{gathered} M_{ij}^{u}(x) = \min_{v=1}^{N_j} M_{ij}^{(u,v)}(x) \\ M_{ij}(x) = \max_{u=1}^{N_i} M_{ij}^{u}(x) \end{gathered} \tag{5}$$

2.2 M^3-GZC Network

Suppose that there are two samples x_i and x_j belonging to class C_i and class C_j, respectively. The Gaussian-zero-crossing function can be adopted to separate these two samples [9], [11] and it is defined as follows:

$$f_{ij}(x) = \exp(-(\frac{|x-c_i|}{\sigma})^2) - \exp(-(\frac{|x-c_j|}{\sigma})^2), \tag{6}$$

where x is the input vector, $\sigma = \lambda|c_i - c_j|$, λ is a constant defined by the user, which actually determines the shape of the GZC function, and λ is empirically set to 0.5 throughout the whole paper.

The output of the M^3-GZC network can be precisely defined as follows:

$$g_i(x) = \begin{cases} 1, & y_i(x) > \theta_i, \\ unknown, & -\theta_j \leqslant y_i(x) \leqslant \theta_i, \\ -1, & -\theta_j > y_i(x). \end{cases} \tag{7}$$

where θ_i and θ_j are the thresholds for C_i and C_j, respectively, and $y_i(x)$ denotes the transfer function of M^3 network for C_i.

2.3 Supervised Clustering Algorithm

For a very large-scale pattern classification problem, supervised clustering is useful as it can effectively reduce the number of training samples. We thus can efficiently train a pattern classifier on the pruned training set.

The ideology of clustering algorithm is as follows. When a new sample comes, group it into the cluster which is nearest to it or let itself to be a new cluster. But it is not so easy to distinguish these two situations. Li and Ye [7,8] defined a grid. Only those samples in the same grid should be in the same cluster. This brings about another problem: how to determine the size of the grid.

Here we follow the concept of receptive field [9,6,11]. The receptive field is determined by the distribution of sample data, and it is the local area around the data. So we can treat the receptive field as the grid. When a new sample comes, we use this receptive field to decide whether it should be assigned to an existing cluster or a new one.

In detail, the clustering process is as follows. Let (x, c) be a new sample. The sample closest to it is marked as (x', c). If it can be covered in the receptive field of (x', c), then these two samples are in the same cluster. Otherwise, (x, c) will be set to a new cluster. And we denote this process as M^3-GZC-C.

Since the receptive field of the samples will overlap, only those samples closest to (x, c) are considered in the above clustering procedure. This may result in a situation in which a cluster center may be covered by the receptive filed of another cluster center. If this case occurs, then these two clusters will be merged. So, we can further compress the number of samples by using this method. According to the discussion above, this processing can be formally addressed as follows. For each cluster (x, c), find its closet neighbor (x', c), if (x, c) can be involved in (x', c), then merge these two clusters. We call this process as M^3-GZC-CC.

3 Experiments

In order to verify the effectiveness of our training sample pruning method, we perform two experiments on Japanese patent classification tasks. In our experiments, we use M^3-GZC-C to prune the training samples, and M^3-GZC-CC to further prune the trining samples until the number of training samples is kept unchanged. We compare the performance of the patent classifiers with and without our training sample pruning method. All of the experiments are run on a PC with Intel Core 2 2.83GHz / 4G RAM.

3.1 Experiment Setup

The data set used in our experiments was collected from the NTCIR-5 patent data set [4] which follows the International Patent Classification (IPC) taxonomy[1]. The IPC is a hierarchically structured system including section, class, subclass, group and subgroup layers. The section layer is the top layer, and subgroup layer is at the bottom. There are about 350,000 new Japanese patents each year, and the patents of year 2001 and 2002 are used in our experiments. A patent document is generally stored in XML format and it is usually consists of three main sections: abstract, claim and description. In our experiments, theses three sections were weighted equally and indexed into a single vector by using the TFIDF algorithm. Then the χ^2_{avg} [12] feature selection method is used. Traditional SVMs and nearest-neighbor algorithm are selected as patent classifiers and compared in our experiments.

3.2 A Two-Class Classification Problem in the Subgroup Layer

Here, we choose the data from the subgroup layer with the categories of H01L021/027 and H01L021/60. After feature selection, the dimension of the data is 1941. The data distribution is shown in Table 1. SVMs with RBF kernel (C=8, g=0.022) are used.

Table 1. Description of training and test data from the subgroup layer

	Training		Test	
	H01L021/027	H01L021/60	H01L021/027	H01L021/60
No. of samples	1256	1003	1045	934

In order to compare the performance of our supervised clustering algorithm with different thresholds, we should search suitable threshold values experimentally. We set θ_i equal to 0.9 and change θ_j from 0.0 to 1, and then evaluate the performance with nearest-neighbor algorithm. The experiment results indicate that θ_i=0.9 and θ_j=0.1 have a good performance with relatively small training sample size. So we use these parameters in the following experiments.

[1] The International Patent Classification, which is commonly referred to as the IPC, is based on an international multi-lateral treaty administered by WIPO. It provides a hierarchical system of language independent symbols for the classification of patents and utility models according to the different areas of technology to which they pertain.

The training and classification results with and without our training sample pruning method are presented in Tables 2 and 3. In Table 2, 'Size ratio' means the ratio of the number of training samples after and before pruning by using the proposed method and 'No. of incorrect outputs' means the number of the test samples classified incorrectly. We can see from this table that the classification accuracy can be improved or at least be kept with a smaller training sample set by using our training sample pruning method.

The results of Table 3 indicate that after clustering the number of support vectors is reduced and the training and test time are thus less than the case without clustering. Why the classification accuracy can be improved? It may be attributed to the fact that the most important support vectors are kept and those less important support vectors are pruned.

Table 2. Performance comparison of the patent classifiers in subgroup layer with and without our training sample pruning method

	M^3-GZC-C		M^3-GZC-CC		Without Clustering	
	NN	SVM	NN	SVM	NN	SVM
No. of Training samples	2164	2164	2136	2136	2259	2259
Size ratio (%)	95.79	95.79	94.56	94.50	100	100
Training time (s)	13.8	0.67	13.5	0.67	14.3	0.69
No. of incorrect output	92	24	91	24	92	25
Classification accuracy (%)	95.57	98.85	95.62	98.85	95.57	98

Table 3. Comparison of the number of support vectors, training time, and test time of the patent classifiers in subgroup layer with and without our training sample pruning method

	M^3-GZC-C	M^3-GZC-CC	Without Clustering
No. of support vectors	239	239	248
Training time (s)	0.67	0.67	0.69
Test time (s)	0.56	0.55	0.61

3.3 An Imbalanced Two-Class Classification Problem in the Section Layer

We choose 84490 samples from the section layer with the categories B and E. These two categories are the most imbalanced ones among eight categories in the section layer. After feature selection, the dimension of the samples is 5000. The data distribution is shown in Table 4. Here we use SVMs with linear kernel.

Table 4. Description of training and test samples from the section layer

	Training		Test	
	B	E	B	E
No. of samples	66991	17499	67359	16896

The training and classification results are presented in Tables 5 and 6. We can see from Table 5 that M3-GZC-CC performs best with the smallest training data set even the sample distribution is greatly imbalanced. The results from Table 6 indicate that after clustering the number of support vector is reduced, so the training and test time are less than the case without clustering.

Table 5. Performance comparison of the patent classifiers in the section layer with and without our training sample pruning method

	M^3-GZC-C		M^3-GZC-P		Without Clustering	
	NN	SVM	NN	SVM	NN	SVM
No. of training samples	75280	75280	70019	70019	84490	84490
Size ratio (%)	89.1	89.1	82.5	82.5	100	100
Training time	751m	38s	750m	23s	864m	49s
Classification accuracy (%)	92.5	97.6	92.5	97.2	92.4	97.6

Table 6. Comparison of the number of support vectors, training time, and test time of the patent classifiers in the section layer with and without our training sample pruning method

	M^3-GZC-C	M^3-GZC-P	Without Clustering
No. of support vectors	11052	6803	11418
Training time (s)	38	23	49
Test time (s)	16	15	17

4 Conclusions

We proposed a training sample pruning method based on a supervised clustering algorithm to deal with large-scale patent classification problems. This method can be used to preprocess training samples before learning. Our preliminary experimental results demonstrate that our method can reduce the number of training samples, while keep or even improve the classification accuracy. Furthermore, both training and test time are decreased due to the smaller size of training sample set. We have also shown that this method can be effectively used for dealing with imbalanced pattern classification problems.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (Grant No. 60903119, Grant No. 60773090 and Grant No. 90820018), the National Basic Research Program of China (Grant No. 2009CB320901), and the National High-Tech Research Program of China (Grant No.2008AA02Z315).

References

1. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Machine Learning 56(1), 89–113 (2004)
2. Eick, C., Zeidat, N., Zhao, Z.: Supervised clustering–algorithms and benefits. In: International Conference on Tools with Artificial Intelligence, pp. 774–776 (2004)
3. Fall, C.J., Benzineb, K.: Literature survey: Issues to be considered in the automatic classification of patents. World Intellectual Property Organization 29 (2002)
4. Fujii, A., Iwayama, M., Kando, N.: Overview of patent retrieval task at NTCIR-5. In: Proceedings of the 5th TCIR Workshop Meeting, pp. 269–277 (2005)
5. Kohonen, T.: Improved versions of learning vector quantization. In: International Joint Conference on Neural Networks, pp. 545–550 (1990)
6. Li, J., Lu, B.: A new supervised clustering algorithm based on min-max modular network with Gaussian-zero-crossing functions. In: International Joint Conference on Neural Networks, pp. 786–793.
7. Li, X., Ye, N.: Grid-and dummy-cluster-based learning of normal and intrusive clusters for computer intrusion detection. Quality and Reliability Engineering International 18(3), 231–242 (2002)
8. Li, X., Ye, N.: A supervised clustering algorithm for computer intrusion detection. Knowledge and Information Systems 8(4), 498–509 (2005)
9. Lu, B., Ichikawa, M.: A Gaussian zero-crossing discriminant function for min-max modular neural networks. In: Knowledge-based Intelligent Information Engineering Systems and Allied Technologies, pp. 298–302 (2001)
10. Lu, B., Ito, M.: Task decomposition and module combination based on class relations: A modular neural network for pattern classification. IEEE Transactions on Neural Networks 10(5), 1244–1256 (1999)
11. Lu, B., Li, J.: A min-max modular network with Gaussian-zero-crossing function. Trends in Neural Computation, 285–313 (2007)
12. Sebastiani, F.: Machine learning in automated text categorization. ACM computing surveys 34(1), 1–47 (2002)