Adaptive Ensemble Learning Strategy Using an Assistant Classifier for Large-Scale Imbalanced Patent Categorization

Qi Kong[1,2], 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
[2] MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems
Shanghai Jiao Tong University, 800 Dongchuan Rd., Shanghai, China, 200240
blu@cs.sjtu.edu.cn

Abstract. Automatic patent classification is of great practical value for saving a lot of resources and manpower. As real patent classification tasks are often very-large scale and serious imbalanced such as patent classification, using traditional pattern classification techniques has shown inefficient and ineffective. In this paper, an adaptive ensemble learning strategy using an assistant classifier is proposed to improve generalization accuracy and the efficiency. The effectiveness of the method is verified on a group of real patent classification tasks which are decomposed in multiple ways by using different algorithms as the assistant classifiers.

1 Introduction

Automatic patent classification is of great importance. Current patent classification mainly relies on human experts while a large-scale of patents are issued annually, e.g., more than 300,000 Japanese patents per year. This circumstance asks for effective automatic processing techniques. However, patent classification is too large a problem to adopt many popular machine learning algorithms.

In detail, patent classification is a large-scale, hierarchical structure and imbalanced text classification task. A lot of works have been done on the task. Previous works mainly focus on a single classifier and don't bring up satisfied results up to now [1]. Typically, such a single classifier may be the state-of-the-art method, SVM, which requires solving a quadratic optimization problem and costing training time that is at least quadratic to the number of training samples. Therefore, even for the efficient training algorithms for SVM such as SMO, large-scale classification problems are still too tough to trickle.

A parallel and modular method named MIN-MAX-modular (M^3), was proposed in [2], in witch a complicated classification problem may be divided into

[⋆] Corresponding author.

K.W. Wong et al. (Eds.): ICONIP 2010, Part I, LNCS 6443, pp. 601–608, 2010.

many smaller independent binary classification problems. Based on MIN-MAX-modular network, Assistant Classifier Learning Strategy (ACLS for short) is proposed in this paper to improve the generalization performance of patent classification and the efficiency as well.

The proposed scheme allows a very large and imbalanced binary classification problem to be divided into independent binary balanced sub-problems. In detail, for the training phase, a large imbalanced training data set will be decomposed into many balanced training subsets and be processed in parallel. Then base learners are trained on all these subsets independently. For each sample in the original training set, the outputs of these base learners are made into vectors and an assistant classifier will learn from the vectors to automatically find an effective ensemble way to output the original class label for each given sample. With all base learners and an assistant ensemble classifier trained, for the recognition phase, an unknown sample is presented to all the base learners; the outputs of all the learners are integrated to make a final solution to the original problem according to the assistant classifier.

The rest of this paper is organized as follows. Section 2 proposes Assistant classifier Learning Strategy and describes its two steps. Section 3 gives the experimental results, which include the comparisons of traditional SVM and ACLS on different decomposition and module combination methods on Japanese patent data set. Section 4 concludes this work.

2 Assistant Classifier Based Module Selection Strategy for Parallel Balanced Learning

Assistant Classifier based Module Selection Strategy (ACMSS) is a novel integration strategy for MIN-MAX-modular classifier (M^3). M^3 classifier is a general framework which is able to solve patent classification problems which is large-scale and imbalanced in a parallel way based on the conquer-and-divide idea.

2.1 Task Decomposition

A large imbalanced real patent training data set is first decomposed into smaller and balanced training sets and parallel processed in the task decomposition phase. The decomposition method has been described in [13,10,3,5].

Two-class classification is one kind of basic classification problem. Many essential classification schemes often start from binary classifier and then adapt to multi-class classifiers. Let $\mathcal{T} = \{(X_l, Y_l)\}_{l=1}^{L}$ be the training set of a K-class classification problem and the K classes are represented by $C_1, C_2, \ldots, C_K$, respectively. $X_l \in R^d$ is the input vector, $Y_l \in R^K$ is the expected output, and L is the number of training samples. Suppose the K training input sets, $\mathcal{X}_1, \ldots, \mathcal{X}_K$ are expressed as $\mathcal{X}_i = \{X_l^i\}_{l=1}^{L_i}$ for $i = 1, \ldots, K$, where L_i is the number of training samples in class C_i, X_l^i is the l-th sample belongs to class C_i and all of $X_l^{(i)} \in \mathcal{X}_i$ have the same expected outputs and $\sum i = 1^K L_i = L$. According to the M^3 network, a K-class problem can be divided into $K \times (K-1)$

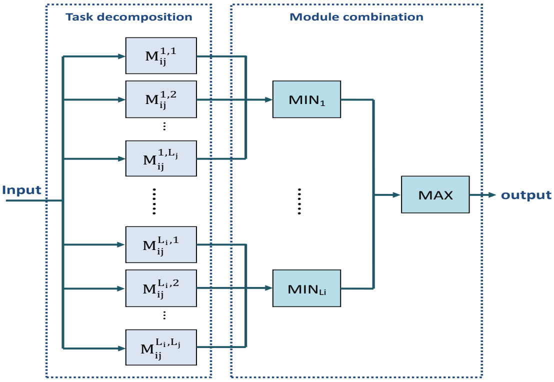

Fig. 1. M^3 contents $L_i \times L_j$ individual network modules, L_i min nodes, and one max node

two-class problems that are trained independently, each of which is given by $\mathcal{T}_{ij} = \{(X_l^{(i)}, +1)\}_{l=1}^{L_i} \cup \{(X_l^{(j)}, -1)\}_{l=1}^{L_j}$, for $i = 1, \ldots, K-1$ and $j = i+1, \ldots, K$. If these two-class problems are still large-scale or imbalanced, they can be further decomposed into relatively smaller two-class problems until suitable for traditional classifiers.

Assume that the input set $\mathcal{X}_i$ is further partitioned into N_i subsets in the form of $\mathcal{X}_{iu} = \{X_l^{(iu)}\}_{l=1}^{L_i^{(u)}}$ for $u = 1, \ldots, N_i$, where L_i^u is the number of training samples included in $\mathcal{X}_{iu}$ and $\cup_u^{N_i} \mathcal{X}_{iu} = \mathcal{X}_i$. After dividing the training input set $\mathcal{X}_i$ into N_i subsets $\mathcal{X}_{iu}$, the training set for each of the smaller and simpler two class problem can be given by $\mathcal{T}_{ij}^{(u,v)} = \{X_l^{(iu),+1}\}_{l=1}^{L_i^{(u)}} \cup \{X_l^{(iv),-1}\}_{l=1}^{L_j^{(v)}}$, for $u = 1, \ldots, N_i, v = 1, \ldots, N_j, i = 1, \ldots, K-1$ *and* $j = i+1, \ldots, K$, where $X_l^{(iu)} \in \mathcal{X}_{iu}$ and $X_l^{(jv)} \in \mathcal{X}_{jv}$ are the input vectors belonging to class C_i and class C_j, respectively, $\sum_{u=1}^{N_i} L_i^{(u)}$ and $\sum_{v=1}^{N_j} L_j^{(v)}$.

After task decomposition, all of the two-class subproblems are treated as completely independent tasks in the learning phase. Therefore, all the two-class sub-problems are efficiently learned in a massively parallel way.

2.2 Module Combination Using Assistant Classifier

After the task decomposition phase, we have trained the modules which are signed to learn associated sub-problems. An unknown sample is presented to all the base learners. Then the outputs of all the learners are integrated to make a final solution to the original problem.

Without losing the generality, if a base classifier gives a positive class prediction for a test sample, it will be denoted “1”, otherwise “0”. For convenience, the outputs of all base classifiers for a certain test sample are illustrated in a dot matrix with base classifiers in each row sharing the same positive training subsets and in each column sharing the same negative training subsets.

With all of the two-class sub-problems learnt by every corresponding base classifiers, all the trained classifiers are integrated into an M^3 classifier. The original integration strategy in [3] is rule-based. In detail, it requires two specific rules, the minimization principle and maximization principle. Figure 1 illustrates an M^3 classifier, where the function of the Min unit is to find a minimum value from its multiple inputs while Max unit is to find a maximum value.

Further combination algorithms or strategies, AMS, SMS and DTMS, have been proposed to improve the original combination strategy for M^3 classifier [11,9,6]. Asymmetric module selection strategy (AMS) is equal to the original MIN-MAX combination but with some heuristic speedup [8]. Symmetric module selection strategy (SMS)[8,12] is not strictly equal to the original MIN-MAX combination any more, though it is still mostly motivated from the latter. In SMS, a pointer is presented to check each output of base classifier from the top-left corner of the matrix and determine the next to be checked base classifier according to the current output class label. The output class label of the original problem is determined by the position where the pointer is finally located. If the pointer will be at the rightmost column, then the output label will be the positive class, otherwise, the negative class. Examples for AMS and SMS are shown in Figure 2(a,b). Decision tree based module selection (DTMS)[6] is further improved from SMS. The different from the latter is that a decision tree is introduce instead to determine which base classifier should be checked. An example of DTMS is shown in Figure 2(c).

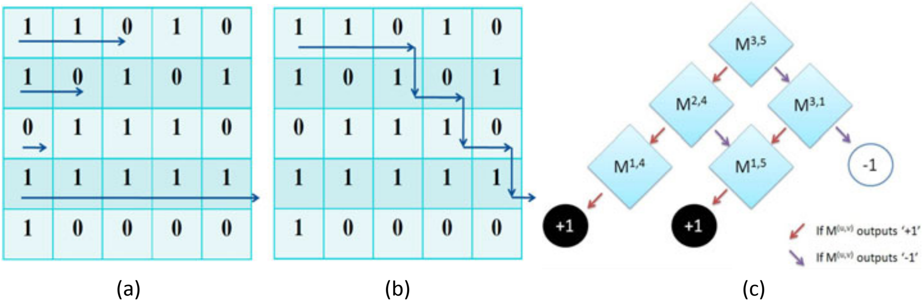

Fig. 2. Classifier selection algorithms: (a) AMS (b) SMS (c) DTMS

However, all the existing integration approaches, either AMS, SMS or DTMS only takes advantage of partial but not all outputs of the base classifiers. For a test sample, some base classifiers do not actually make any contribution for the final output. In addition, all these approaches should work on concrete classifiers whose output should be either 0 or 1, instead of continuous confidence values in the MIN-MAX module combination. ACMSS is thus proposed to overcome the above drawbacks of the existing methods.

ACMSS works in the following way. For the training phase, For each sample in the original training set, the outputs of those base learners are made into vectors (shown in Figure 3). Then an assistant classifier will learn from the vectors to find an effective ensemble way automatically. With all base learners and an assistant ensemble classifier trained, for the recognition phase, an unknown sample is presented to all the base learners; the outputs of all the learners are integrated to make a final solution to the original problem according to the assistant classifier.

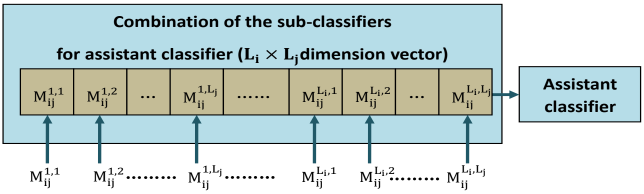

Fig. 3. Assistant classifier learns from the vectors composed by the outputs of $L_i \times L_j$ modules

ACMSS can be considered to acquire and absorb the knowledge which has been learnt by many weak base classifiers. We introduce ACMSS to compute the weights of base classifiers automatically to get a better performance on large-scale imbalanced data. We note that meta-learning in [4] is based on the combine of different classification algorithms while ACMSS is based on task decomposition and base classifiers' combination. The training set size of each base classifier of ACMSS is much smaller than those of meta-learning.

3 Experiments

The data sets used in our experiments are from the NTCIR-8 patent mining shared task which follows the International Patent Classification (IPC) taxonomy. Patent data set is large-scale, hierarchical and imbalanced.

The patent data of year 2002 are adopted in our experiments. The $\mathcal{X}_{avg}$ feature selection method is used for vector space model construction. And the patents are indexed into a single vector with 5000 dimensional by using the $tfidf$ algorithm. SVM are selected as the base classifier and the assistant classifier with linear kernel. $\mathcal{C}$ (the trade-off between training error and margin) is set to be $\frac{1}{avg.\|x\|^2}$ where x is the samples of the subset.

Data set is decomposed with various strategies according to [13]. These strategies include random(R-), hyperplane(H-) [10], centroid connection(C-) [7] and year and category based prior knowledge (YC-). The main idea of hyperplane and centroid connection decomposition methods is as shown in Figure 4. The prior knowledge decomposition method divides the date set by the information of the release data and IPC label of each patent.

We choose a subset of these Japanese patents of year 2002. The data contains 150,000 samples including 75,000 samples of section G (Physics patents) and 75,000 samples of section H (Electricity patents). There are 100,000 samples

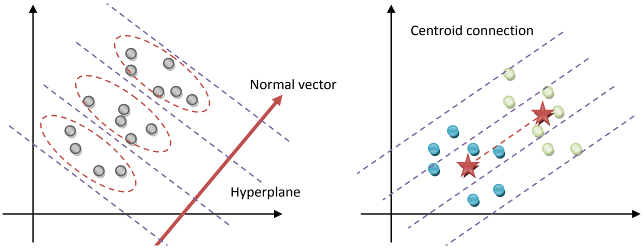

Fig. 4. Left is the hyperplane classification method. hyperplane is orthogonal to the normal vector which is produced by PCA. Right is the centroid connection method. The stars are the centroid of different categories. Samples are decomposed according to the distance to the centroid connection line.

Table 1. The experimental results of balanced and imbalanced patent data

	balanced data			imbalanced data		
	Precision%	Recall%	Macro-F_1%	Precision%	Recall%	Macro-F_1%
T-SVM	89.45	89.36	89.40	97.37	61.51	79.92
R-AMS	**89.47**	88.28	88.87	82.17	93.75	87.58
R-SMS	**89.47**	88.28	88.87	82.17	93.75	87.58
R-DTMS	88.65	88.42	88.53	83.24	**93.78**	88.20
R-ACMSS	89.27	**90.16**	**89.71**	**88.37**	88.80	**88.58**
H-AMS	**88.32**	87.13	87.72	81.21	**92.77**	86.61
H-SMS	**88.32**	87.13	87.72	81.21	**92.77**	86.61
H-DTMS	87.89	87.56	87.72	82.29	92.69	87.18
H-ACMSS	88.31	**89.05**	**88.68**	**87.39**	87.98	**87.68**
CC-AMS	89.46	88.33	88.89	82.18	**93.73**	87.58
CC-SMS	**89.47**	88.32	88.89	82.18	**93.73**	87.58
CC-DTMS	88.91	88.45	88.68	83.31	93.61	88.16
CC-ACMSS	89.37	**90.16**	**89.76**	**88.47**	89.08	**88.77**
YC-AMS	90.13	89.19	89.66	83.09	94.27	88.33
YC-SMS	90.13	89.13	89.66	83.09	94.27	88.33
YC-DTMS	89.94	89.52	89.73	84.35	**94.59**	89.18
YC-ACMSS	**90.39**	**91.15**	**90.77**	**89.51**	90.08	**89.79**

note: Decompose methods: R-Randomly, H-Hyperplane, CC-Centroid Connection, YC-priori knowledge(Year & Category) decomposition. Combination methods: AMS-asymmetric selection, SMS-symmetric selection. DTMS employs C4.5. The parameter $\mathcal{C}$ of ACMSS on balanced data is 0.02, on imbalanced data is 0.025.

for the training and 50,000 for the test. Training data are decomposed into 25×25 sub-classification problems with 2,000 positive samples and 2,000 negative samples in each subset.

Similar with the above balanced patent data, we choose a subset of Japanese patents of year 2002 as an imbalanced data set. It contains 110,000 samples including 80,000 samples for training and 30,000 for test. Training set has 10,000 samples of section G and 70,000 samples of section H. Training data are decomposed into 5 × 35 sub-classification problems with 2,000 positive samples and 2,000 negative samples in each subset.

Four module selection methods (AMS, SMS, DTMS and ACMSS) are used for combination. The experimental results are shown in Table 1.

From the experimental results, we can see that the ACMSS, using SVM as the assistant classifier, yields the best performance in most cases, especially with prior knowledge based decomposition.

Either AMS, SMS or DTMS only takes advantage of partial but not all outputs of the base classifiers. For a test sample, some base classifiers do not actually make any contribution for the final output. ACMSS can be considered to acquire and absorb the knowledge which has been learnt by weak base classifiers. In addition, ACMSS works on concrete classifiers whose output are continuous confidence values, instead of either 0 or 1 in AMS, SMS or DTMS. ACMSS computes the weights of base classifiers automatically to get a better performance on large-scale data.

4 Conclusion

Current patent classification mainly relies on human experts while a large-scale of patents are issued annually. This circumstance asks for effective automatic processing techniques. However, patent classification is too large and imbalanced a problem to adopt many popular machine learning algorithms. The problem is too tough to trickle for a single classifier and the imbalance of data set seriously affects the classification results.

Taking into account this difficulty, the focus is on seeking a parallel classification algorithm. M^3 classifier is a general framework which is able to solve patent classification problems which is large-scale and imbalanced in a parallel way based on the conquer-and-divide idea. Complicated classification problem may be divided into many smaller balanced independent binary classification problems. With the popularity of distributed computing, the parallel learning strategy is showing strong advantages for solving practical problems.

ACMSS is proposed in this paper to improve the generalization performance and the efficiency. ACMSS computes the weights of base classifiers automatically to get a better performance on large-scale imbalanced data. ACMSS works on concrete classifiers whose output are continuous confidence values, instead of either 0 or 1 in AMS, SMS or DTMS. ACMSS can be considered to absorb the knowledge learnt by weak base classifiers employing an assistant classifier.

This research enables us to know further about large-scale imbalanced patent categorization strategies, especially the parallel methods based on the conquer-and-divide idea. ACMSS is of better adaptive ability and strong generalization since many classifier algorithm can be employed as the assistant classifier. The adaptive ensemble learning strategy that we propose performs the best in solving large-scale both balanced and imbalanced categorization problems as shown in the experiments, which is valuable for real-world applications.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (Grant No. 60773090 and Grant No. 90820018), the National Basic Research Program of China (Grant No. 2009CB320901), the National High-Tech Research Program of China (Grant No. 2008AA02Z315), and the Science and Technology Commission of Shanghai Municipality (Grant No. 09511502400).

References

1. Fujino, A., Isozaki, H.: Multi-label classification using logistic regression models for NTCIR-7 patent mining task. In: Proceedings of NTCIR-7 Workshop Meeting, Tokyo, Japan (2008)
2. Lu, B.L., Bai, Y., Kita, H., Nishikawa, Y.: An efficient multilayer quadratic perceptron for pattern classification and function approximation. In: Proceeding of International Joint Conference on Neural Networks, pp. 1385–1388 (1993)
3. Lu, B.L., 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)
4. Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. Artificial Intelligence Review 18(2)
5. Wang, K., Zhao, H., Lu, B.L.: Task decomposition using geometric relation for min-max modular svms. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3496, pp. 887–892. Springer, Heidelberg (2005)
6. Wang, Y., Lu, B.L., Ye, Z.F.: Module combination based on decision tree in min-max modular network. In: Proceedings of International Conference on Neural Computation, Madeira, Portugal, pp. 555–558 (2009)
7. Ye, Z.F.: Parallel min-max modular support vector machine with application to patent classification (in chinese). Master Thesis, Shanghai Jiao Tong University (2009)
8. Zhao, H., Lu, B.L.: Improvement on response performance of min-max modular classifier by symmetric module selection. In: Proceeding of Second International Symposium Neural Networks, Chongqing, China, vol. 3971, pp. 39–44 (2005)
9. Zhao, H., Lu, B.: Analysis of fault tolerance of a combining classifier. In: Proceedings of 12th International Conference on Neural Information, Taipei, Taiwan, pp. 888–893 (2004)
10. Zhao, H., Lu, B.: Determination of hyperplane by pca for dividing training data set. In: Proceeding of 12th International Conference on Neural Information, Taipei, Taiwan, vol. 3173, pp. 755–760 (2005)
11. Zhao, H., Lu, B.: A general procedure for combining binary classifiers and its performance analysis. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3610, pp. 303–312. Springer, Heidelberg (2005)
12. Zhao, H., Lu, B.: A modular reduction method for k-NN algorithm with self-recombination learning. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3971, pp. 537–542. Springer, Heidelberg (2006)
13. Zhao, H., Lu, B., Wen, Y.M., Wang, K.A.: On effective decomposition of training data sets for min-max modular classifier. In: Proceeding of 12th International Conference on Neural Information, Taipei, Taiwan, pp. 343–348 (2005)