Incorporating Prior Knowledge into Task Decomposition for Large-Scale Patent Classification

Chao Ma[1], Bao-Liang Lu[1,2,⋆], and Masao Utiyama[3]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] MOE-Microsoft Key Lab for Intelligent Computing and Intelligent System
Shanghai Jiao Tong University
800 Dong Chuan Road, 200240, Shanghai, China
bllu@sjtu.edu.cn
[3] National Institute of Information and Communications Technology (NICT)
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0288 Japan
mutiyama@nict.go.jp

Abstract. With the adoption of min-max-modular support vector machines (SVMs) to solve large-scale patent classification problems, a novel, simple method for incorporating prior knowledge into task decomposition is proposed and investigated. Two kinds of prior knowledge described in patent texts are considered: time information, and hierarchical structure information. Through experiments using the NTCIR-5 Japanese patent database, patents are found to have time-varying features that considerably affect classification. The experimental results demonstrate that applying min-max modular SVMs with the proposed method gives performance superior to that of conventional SVMs in terms of training time, generalization accuracy, and scalability.

1 Introduction

In the modern world, patents and patent applications are important factors in measuring the levels and capabilities of scientific and technological progress of a country or company. Automatic patent classification is a multi-class problem with the following characteristics distinguishing it from traditional pattern classification problems:(1) It is a very large-scale problem in terms of both the number of training samples and the number of categories. (2) It is a typical hierarchical pattern classification problem. (3) Training samples collected from different years have time-varying characteristics. (4) The number of available training samples continuously increases. (5) It is a multi-label problem.

Because of its great importance in patent analysis and patent mining, automatic patent classification has received much attention in recent years [1,2,3,4,5,6]. Many patent classifiers have been studied, such as the naive Bayes classifier, k-NN, support vector machines (SVMs), neural networks, and decision rules. Fall *et al.* suggested that SVMs are suitable patent classifiers capable of achieving the best performance among

⋆ Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 784–793, 2009.

these classifiers [5,6]. SVMs suffer, however, from time and space complexities for large-scale patent classification problems. To scale SVMs up to large-scale pattern classification problems, Lu *et al.* proposed a min-max modular support vector machine (M^3-SVM) [7]. The basic idea behind M^3-SVMs is to apply the divide-and-conquer strategy: decomposing a complex problem into a series of simple subproblems; learning all of the subproblems by using SVMs; and integrating the trained SVMs according to the minimization and maximization principles [8].

In this paper, we adopt M^3-SVMs for large-scale patent classification problems and propose a novel method for incorporating prior knowledge into task decomposition. We focus on two kinds of prior knowledge described in patent texts: time information, and hierarchical structure information. To sufficiently utilize the time information, we explore the relationship between patent texts and classification performance. Examining the NTCIR-5 Japanese patent database [9], which consists of more than two million unexamined Japanese patent applications, we find that patent applications have time-varying features, and that this time-dependence property considerably affects learning and classification. we compare the proposed method with the existing task decomposition approaches. The experimental results demonstrate that applying M^3-SVMs with the proposed task decomposition method achieves performance superior to that of conventional SVMs, in terms of both training time and generalization accuracy.

The rest of the paper is organized as follows. The patent classification problem is described in section 2. In section 3, the time-varying features of patent texts are analyzed. Section 4 briefly introduces M^3-SVMs, and section 5 proposes our effective task decomposition strategy based on prior knowledge. The experiments and results are presented in section 6, and our conclusions are given in section 7.

2 Problem Description

We address the task of Japanese patent classification on the NTCIR-5 patent database. The NTCIR-5 database adopts the International Patent Classification (IPC) taxonomy, which provides a common classification scheme for patents and inventions. The IPC is a hierarchically structured system consisting of five levels: section, class, subclass, group,

Table 1. Number of patents in eight section categories, 1993-1999

Section	1993	1994	1995	1996	1997	1998	1999	Total
A	30,583	31,316	28,357	25,444	22,475	32,427	33,126	203,728
B	65,538	68,474	68,130	68,278	62,436	68,148	69,648	470,652
C	30,747	31,834	34,163	37,996	35,700	31,198	31,494	233,132
D	4,904	5,228	5,794	6,127	5,604	4,642	4,968	37,267
E	18,605	18,000	16,114	13,690	11,099	18,604	18,810	114,922
F	30,296	31,188	29,358	28,258	26,671	31,403	32,938	210,112
G	77,692	81,691	81,677	88,716	95,679	79,158	83,942	588,555
H	72,589	72,164	72,544	81,486	86,834	75,305	80,594	541,516
Total	330,954	339,895	336,137	349,995	346,498	340,885	355,520	2,399,884

and subgroup. The top level is the section level, which contains eight categories labeled from 'A' through 'H'. The second level is the class level, which contains 120 categories expressed by two digits after the section label, such as 'A01'. The third level is the subclass level, which has 615 categories represented by a capital letter following the class label, such as 'A01B'. The fourth and fifth levels are the group level and subgroup level, respectively. In general, current research is mainly concentrated on the top three levels, because the definitions of the group and subgroup levels are frequently changed.

All of the unexamined Japanese patent applications published from 1993 through 1999 in the NTCIR-5 patent database were used in this study. Table 1 summarizes the distribution of these patent applications. From the table, we can see that the total number of patent applications in this period was nearly 2.4 million. A patent text consists of four parts: *Abstract*, *Claim*, *Description*, and other descriptive information, such as *Title* and *IPC* labels.

3 Time-Varying Features of Patents

While patent classification techniques such as feature extraction methods and classification algorithms have been extensively studied, the time-varying features of patents issued in different periods and their influence on classification have not been explored yet. In this section, we address these issues.

One unique characteristic of patents is their time dependence. On the one hand, the words used by people change over time. This is the evolution of language usage over time: what people talk about, and what vocabulary they use. Many other data sets, such as web logs (blogs) [10], have this same characteristic as patents. On the other hand, technical directions also change with time. Therefore, we suspect that as the time interval between the training data and test data decreases, the more similar their distribution becomes.

3.1 Influence on Classification

To investigate how training data collected from different periods affect the generalization performance of a patent classifier, we constructed seven training data sets and one test data set by using the NTCIR-5 patent database. The training data sets consisted of all of the patent applications published in each year from 1993 to 1998, while the test data set consisted of all of the patent applications of 1999. Two popular classification methods, SVMs and the k-NN algorithm, were used as classifiers. We used SVM^{light} to train the SVMs with a linear kernel. Figure 1 shows the experimental results. The changing tendency of classification performance seen in the figure demonstrates that as the time interval between the training data and test data decreases, the more the classification performance of the patent classifier improves.

3.2 Variations of Different Words

To explain classification performance tendency observed in Figure 1, we performed a simple analysis of different words and their frequencies in the training data and test data. To facilitate description, we give some definitions of terminology used below.

Word only in test (WOT): a word that appears only in the test data set.
Sum of WOT frequencies (SWF): the total frequency of all WOTs.
Average frequency of WOTs (AFW): the average frequency of all WOTs.

We counted the number of different WOTs and found an interesting phenomenon, illustrated in Figure 2. As the time interval between the training data and test data decreases, the number of WOTs decreases, and the value of the SWF also decreases. In other words, the number of common words increases over time. This confirms the time-varying feature of patent text.

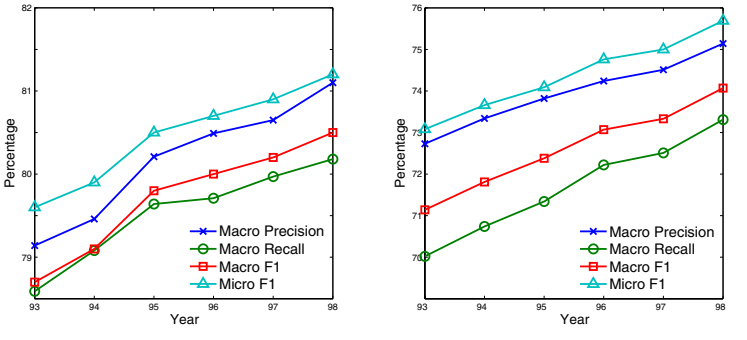


Fig. 1. Changing tendency of classification performance with different time intervals between the training data and test data: (a) SVMs with a linear kernel; and (b) kNN algorithm

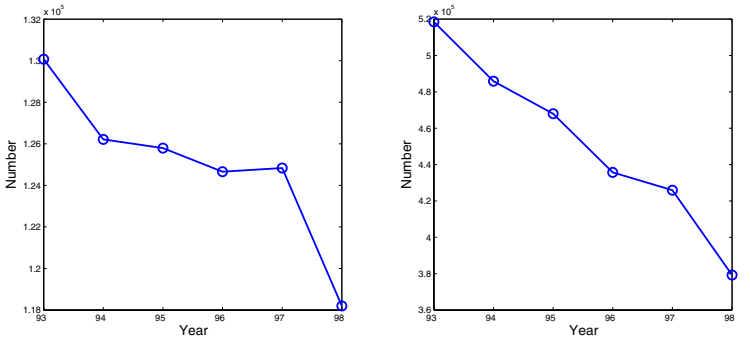
3.3 Variations of Word Frequency

The average frequency of words appearing only in the test set is about 3.6, while the average word frequency in each training data set is about 925. This means that the words only appearing in the test set are keywords, such as special field-specific words. We found that the average percentage of words appearing only in the test set is 52%, much larger than their frequency of 0.21%. This indicates that these words are domain-dependent words. We also found that when the time interval between the training set and the test set is smaller, the frequency of words appearing only in the test set decreases, as listed in Table 2. This indicates that more recent training data sets contain common words that do not appear in their previous years' data sets. In other words, the habits of using words change with time.

These time-varying features of patents are used in our proposed task decomposition method as a kind of prior knowledge, as explained in section 5. The importance of incorporating this knowledge is demonstrated by our experiments.

4 Min-Max Modular SVM

In this section, we briefly introduce M^3-SVMs [8]. The working procedure of M^3-SVMs includes three main steps: task decomposition, SVM training, and module combination.



(a) (b)

Fig. 2. Statistics of different words and their frequencies: (a) number of different words appearing only in the test set; and (b) sum of frequencies for words appearing only in the test set

Table 2. Changing tendency of the AFW with time

Year	93	94	95	96	97	98
AFW	4.0	3.9	3.7	3.5	3.4	3.2

4.1 Task Decomposition

Before training M^3-SVMs, a K-class problem should be divided into $K(K-1)/2$ two-class subproblems by a one-versus-one strategy. Let $\mathcal{T}_{ij}$ be the given training data set for a two-class classification problem:

$$\mathcal{T}_{ij} = \{(X_l^{(i)}, +1)\}_{l=1}^{L_i} \cup \{(X_l^{(j)}, -1)\}_{l=1}^{L_j} \quad \text{for } i = 1, \cdots, K \text{ and } j = i+1, \cdots, K, \tag{1}$$

where $X_l^{(i)} \in \mathcal{X}_i$ and $X_l^{(j)} \in \mathcal{X}_j$ are the training inputs belonging to classes $\mathcal{C}_i$ and $\mathcal{C}_j$, respectively, $\mathcal{X}_i$ is the set of training inputs belonging to class $\mathcal{C}_i$, L_i denotes the number of data in $\mathcal{X}_i$,

Assume that $\mathcal{X}_i$ is partitioned into N_i subsets in the form

$$\mathcal{X}_{ij} = \{X_l^{(ij)}\}_{l=1}^{L_i^{(j)}} \quad \text{for } j = 1, \cdots, N_i \text{ and } i = 1, \cdots, K, \tag{2}$$

where $1 \leq N_i \leq L_i$ and $\cup_{j=1}^{Ni} \mathcal{X}_{ij} = \mathcal{X}_i$.

After partitioning $\mathcal{X}_i$ into N_i subsets, every two-class subproblem $\mathcal{T}_{ij}$ defined by Eq. (1) can be further divided into $N_i \times N_j$ relatively smaller and more balanced two-class subproblems, as follows:

$$\mathcal{T}_{ij}^{(u,v)} = \{(X_l^{(iu)}, +1)\}_{l=1}^{L_i^{(u)}} \cup \{(X_l^{(jv)}, -1)\}_{l=1}^{L_j^{(v)}} \quad \text{for } u = 1, \cdots, N_i, \ v = 1, \cdots, N_j, \ i = 1, \cdots, K, \text{ and } j = i+1, \cdots, K, \tag{3}$$

where $X_l^{(iu)} \in \mathcal{X}_{iu}$ and $X_l^{(jv)} \in \mathcal{X}_{jv}$ are the training inputs belonging to classes $\mathcal{C}_i$ and $\mathcal{C}_j$, respectively, $\sum_{u=1}^{N_i} L_i^{(u)} = L_i$, and $\sum_{v=1}^{N_j} L_j^{(v)} = L_j$.

4.2 SVM Training

In the learning phase, each of the two-class subproblems can be treated as a completely independent, non-communicating problem. Therefore, all the two-class subproblems defined by Eq. (3) can be efficiently learned in a serial or massively parallel way.

From Eqs. (1) and (3), we see that a K-class problem is divided into

$$\sum_{i=1}^{K-1} \sum_{j=i+1}^{K} N_i \times N_j \tag{4}$$

two-class subproblems. The number of training data for each of the two-class subproblems is about

$$\lceil L_i/N_i \rceil + \lceil L_j/N_j \rceil, \tag{5}$$

Since $\lceil L_i/N_i \rceil + \lceil L_j/N_j \rceil$ is independent of the number of classes K, the size of each of the two-class subproblems is much smaller than the original K-class problem for reasonable N_i and N_j.

4.3 Module Combination

After every individual SVM is successfully trained on the corresponding two-class subproblem, all of the trained SVMs are integrated into an M^3-SVM with MIN and MAX units according to two combination principles: the minimization principle, and the maximization principle [8]. The function of the MIN unit is to find a minimum value from its multiple inputs, while the function of the MAX unit is to find a maximum value from its multiple inputs.

5 Incorporating Prior Knowledge

"When everything fails, ask for additional domain knowledge" is the current motto of machine learning. Various previous works have demonstrated that incorporating prior knowledge can considerably improve the performance of learning systems [11]. In this section, we present a novel method for incorporating prior knowledge into task decomposition of M^3-SVMs. We explore two kind of prior knowledge: time information, and hierarchical structure information.

We consider the problem of classifying the eight section-level categories in the NTCIR patent database. Suppose that all of the patent texts from 1993 to 1997 are used as training data ($\mathcal{S}_5$), and the rest, from 1998 to 1999, is used as testing data. From Table 1, we see that the numbers of training and test data are 2,044,364 and 696,405, respectively. If we apply a one-versus-one strategy, the original eight-class patent classification problem is divided into 28 two-class subproblems. According to Eq. 2, among these 28 two-class subproblems, the largest is $\mathcal{T}_{\mathrm{G,H}}$, which has 811,072 training data.

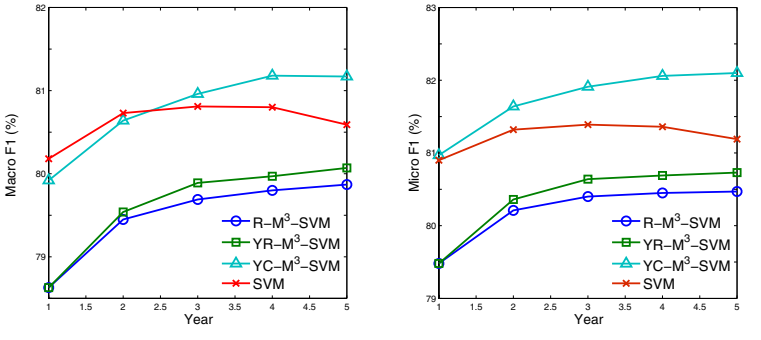
Although the number of training data in $\mathcal{T}_{\mathrm{G,H}}$ is much smaller than in the original problem, this problem is still large scale and difficult to solve.

One of the most important advantages of M^3-SVMs over traditional SVMs is that a large-scale, two-class subproblem can be further divided into a series of two-class subproblems according to Eq. 3. Since each subset defined by Eq 2 represents a local distribution of the entire training data set in the feature space, after training all of the two-class subproblems, the combined results represent the local distribution more accurately. Therefore, the most important factor is that each subset must represent the corresponding local distribution. In order to divide classes into subsets representing local distributions, we use prior knowledge of the time and hierarchical structure, because we have verified that patents close to each other in time are also close to each other in the distribution of the feature space.

On the other hand, patent applications classified by human experts into the same category are naturally close to each other semantically, and therefore, they should be close to each other in the feature space. According to this observation, we first divide patent texts in the same section category into a series of subsets by year. For example, all of the training data belonging to 'A' in $\mathcal{S}_5$ are divided into five subsets by year. That is, the five subsets consist of 30,583, 31,316, 28,357, 25,444, and 22,475 patents.

6 Experiments and Results

To evaluate the effectiveness of applying M^3-SVMs with our proposed task decomposition method for large-scale patent classification, we carried out experiments on the NTCIR-5 Japanese patent database.



(a) (b)

Fig. 3. Performance comparison of four different classifiers: (a) macro F1, and (b) micro F1

6.1 Experimental Settings

We adopted a hierarchical text classification model and focused on the problem of classifying the eight categories at the section level. Note that a patent has one main category label and can also have several compensatory labels. In the experiments, we simplified

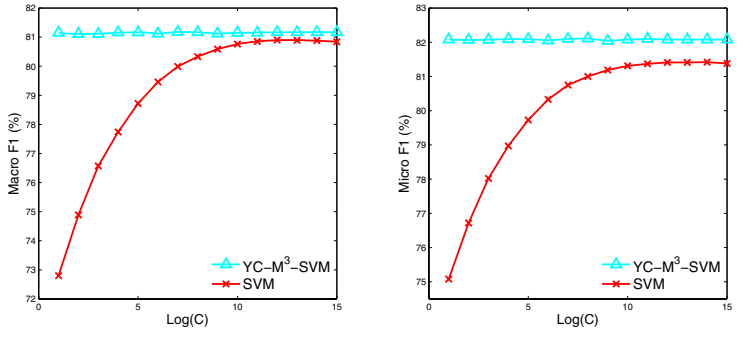
the multi-label problem to a unique-label problem by only considering the main label of every patent.

Four classification methods were used in the experiments: conventional SVMs, and M^3-SVMs with three different task decomposition strategies. All of the experiments were performed on a Lenovo cluster system consisting of three fat nodes and thirty thin nodes. Each fat node had 32 GB of RAM and two 3.2-GHz quad-core CPUs, while each thin node had 8 GB of RAM and two 2.0-GHz quad-core CPUs. Experiments with the conventional SVMs were performed on the fat nodes, while experiments with the M^3-SVMs were done on the thin nodes. We also used the following settings:

1) SVMs with a linear kernel were trained by SVM^{light} [12]. These SVMs acted as a baseline algorithm and component classifiers for the M^3-SVMs.
2) We used all of the patents from 1998 to 1999 as test data and constructed five different training data sets. The training data sets contained all of the patent applications published in the following time periods: 1997, from 1996 to 1997, from 1995 to 1997, from 1994 to 1997, and from 1993 to 1997.
3) We selected 5000 features by using the χ_{avg} algorithm [12]. We had previously varied the number of features from 2500 to 160,000 and found that 5000 was the smallest number giving nearly top performance.
4) For a random decomposition strategy, we set the subset size to 2000 by considering both the accuracy and the time cost of our experiments.
5) We introduced task decomposition methods based on two different kinds of prior knowledge. The first method used only time information, and the classifiers were called YR-M^3-SVMs. The second method used both time information and hierarchical structure information, and the classifiers were called YC-M^3-SVMs.

6.2 Results

Figure 3 shows the classification performance of the SVMs and M^3-SVMs. From this figure, we can see that the two task decomposition methods based on prior knowledge outperformed the random strategy. The YC-M^3-SVMs achieved the best accuracy,



(a) (b)

Fig. 4. Performance variation with changes in the training parameter, C: (a) macro F1, and (b) micro F1

superior to that with the traditional SVMs. These results show that applying prior knowledge improved the performance of the M^3-SVMs.

Another interesting phenomenon can be observed in Figure 3. When the number of training data was increased, the performance of the M^3-SVMs became better and better, surpassing that of the conventional SVMs, which dropped.

The training time for the M^3-SVMs could be reduced to 10% of that for the traditional SVMs, which is much faster. Though the response time of the M^3-SVMs was longer than that of the SVMs, they could classify one patent within 2 ms.

Another interesting factor is parameter tuning. Figure 4 shows the relationship between the classification performance and the value of the penalty parameter C. We can see that the M^3-SVMs were very robust with respect to C, because two-class subproblems are simple and almost linearly separable. In contrast, C was an important parameter for the traditional SVMs, which could achieve their best performance only when C was sufficiently large. As a result, using SVMs should be very time consuming.

7 Conclusions

In this paper, we have described the time-varying features of patent texts and have proposed a novel method for incorporating prior knowledge into task decomposition for M^3-SVMs The results of our experiments on the NTCIR-5 patent database demonstrated that applying our method with M^3-SVMs enabled them to easily incorporate time and hierarchical structure information into learning. This resulted in performance superior to that of random task decomposition and traditional SVMs, which do not consider any prior knowledge during learning. The lower time cost of our parallel system is important for training on large data sets. The conclusions that we obtained here can be generalized to other data sets with the same characteristics as those of patent data.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China via the grants NSFC 60773090, and the Fujitsu Research and Development Center Co., Ltd., Beijing, China.

References

1. Krier, M., Zacc, F.: Automatic categorisation applications at the european patent office. World Patent Information 24, 187–196 (2002)
2. Larkey, L.S.: Some issues in the automatic classification of us patents. In: Learning for Text Categorization, pp. 87–90. ACM, New York (1998)
3. Larkey, L.S.: A patent search and classification system. In: ACM DL 1999, pp. 179–187. ACM, New York (1999)
4. Mase, H., Tsuji, H., Kinukawa, H., Ishihara, M.: Automatic patents categorization and its evaluation. Information Processing Society of Japan Journal 39 (1998)
5. Fall, C.J., Benzineb, K.: Literature survey: Issues to be considered in the automatic classification of patents. World Intellectual Property Organization 29 (2002)
6. Fall, C.J., Törcsvári, A., Benzineb, K., Karetka, G.: Automated categorization in the international patent classification. SIGIR Forum 37(1), 10–25 (2003)

7. Lu, B.L., Wang, K.A., Utiyama, M., Isahara, H.: A part-versus-part method for massively parallel training of support vector machines. In: IJCNN, pp. 735–740. IEEE Press, New York (2004)
8. 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), 1244–1256 (1999)
9. Iwayama, M., Fujii, A., Kando, N.: Overview of classification subtask at ntcir-5 patent retrieval task. In: NTCIR-5 Workshop Meeting (2005)
10. Glance, N., Hurst, M., Tomokiyo, T.: Blogpulse: Automated trend discovery for weblogs. In: WWW 2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics (2004)
11. Chang, M.W., Ratinov, L.A., Roth, D.: Guiding semi-supervision with constraint-driven learning. In: ACL (2007)
12. Joachims, T.: Making large-scale svm learning practical. Kernel Methods-Support Vector Learning (1999)