Front. Comput. Sci. China 2009, 3(1): 109–122
DOI 10.1007/s11704-009-0013-7

RESEARCH ARTICLE

Incorporating prior knowledge into learning by dividing training data

Baoliang LU(✉)[1,2], Xiaolin WANG[1], Masao UTIYAMA[3]

1 Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
2 MOE-Microsoft Key Lab for Intelligent Computing and Intelligent Systems, Shanghai Jiao Tong University, Shanghai 200240, China
3 National Institute of Information and Communications Technology (NICT), Kyoto 619-0288, Japan

Abstract In most large-scale real-world pattern classification problems, there is always some explicit information besides given training data, namely prior knowledge, with which the training data are organized. In this paper, we proposed a framework for incorporating this kind of prior knowledge into the training of min-max modular (M^3) classifier to improve learning performance. In order to evaluate the proposed method, we perform experiments on a large-scale Japanese patent classification problem and consider two kinds of prior knowledge included in patent documents: patent's publishing date and the hierarchical structure of patent classification system. In the experiments, traditional support vector machine (SVM) and M^3-SVM without prior knowledge are adopted as baseline classifiers. Experimental results demonstrate that the proposed method is superior to the baseline classifiers in terms of training cost and generalization accuracy. Moreover, M^3-SVM with prior knowledge is found to be much more robust than traditional support vector machine to noisy dated patent samples, which is crucial for incremental learning.

Keywords prior knowledge, patent classification, support vector machine, min-max modular network, task decomposition

1 Introduction

The automated classification has witnessed a new trend in the last few years. The previous dorminant way of building a classifier purely from labeled samples, namely training data set, turns out to be less attractive since it seems to have reached a plateau in accuracy. To build more accurate classifiers, many researchers are now trying to incorporate prior knowledge into learning.

A variety of methods have been reported on this topic. Overall these methods differ widely as they are tightly bound to the forms of the prior knowledge and the classification tasks. Liu et al. [1] worked on the text classification task of 20 New Groups†, and they adopted a combined approach of estimation maximum and naive Bayes to build a classifier from prior knowledge, which is in the form of key words for target classes. Wu and Srihari [2] transferred prior knowledge into the weights of each training data point, and the classifier of weighted margin support vector machine was applied on this enhanced training data set. Schapire et al. [3] worked on audio classification, and they emerged the prior knowledge, which were some rough rules of classifying samples proposed by domain experts, into the boosting algorithm. Zhu and Chen [4] took a domain dictionary as prior knowledge in classifying Chinese documents. If certain term appeared in the document, the explanation of this term in the dictionary was also taken into consideration when classifying this document by means of some mathematical formula. Dayanik et al. [5] used the prior knowledge of the target classes, which was in the form of textual descriptions, while working on several well-known benchmarks of text classification, including TREC 2004, Reuters-21578 and RCV1. In their method, terms mentioned by prior knowledge were endowed priority over the rest to be powerful features.

Received June 29, 2008; accepted November 25, 2008

E-mail: bllu@sjtu.edu.cn, mutiyama@nict.go.jp

† http://www.cs.cmu.edu/afs/cs/project/theo-11/naive-bayes/20_newsgroups.tar.gz

Min-max modular (M^3) network is an ensemble learning approach proposed by Lu and Ito [6,7]. The intuition of this method is to handle complex classification problem by divide-and-conquer strategy, that is, decomposing a complex problem into a series of simple subproblems, and using one classifier for each subproblem. Meanwhile, this method has the merit of speeding up the training of the classifiers since subproblems can be processed in parallel, which is precious for real-world applications. Though theoretically an M^3-classifier can solve any complex classification problem, yet it still reach a plateau of accuracy like many other classification methods in practical use, partly due to the generalization error. Therefore, some other ways still need to be tried so as to improve it, such as incorporating prior knowledge into learning.

Now let us explain how we come to the idea of incorporating prior knowledge into dividing the training data. On one side, how to decompose the data set of a class for an M^3-classifier is not perfectly solved so far. On the other side, there is always some extra information, namely prior knowledge, for most large-scale pattern classification problems, with which the training data are organized. We merge these two sides of consideration, and find out that such prior knowledge about the training data is actually precious clue to decompose the data set. Thus a new decomposition method with the prior knowledge incorporated into is composed.

In this paper, the experimental demonstration is performed on a database of Japanese patent documents, and two kinds of information are taken as prior knowledge: publishing date and labels' hierarchical structure. However, our method can be applied to any other data set, only if it is well organized by some extra information.

The rest of this paper is organized as follows. The necessary background knowledge are presented at first in the next two sections, Section 2 for M^3-classifier and Section 3 for patent classification task. Then the following Section 4 describes our method of incorporating prior knowledge into task decomposition in detail. After that, Sections 5 and 6 present the experimental settings and results, and Section 7 presents some further analysis on these results. At last we give the conclusions in Section 8.

2 Min-max modular network

The M^3-network was first introduced by Lu and Ito [6]. Before training this network, a K-class problem needs to be divided into K two-class subproblems by one-against-rest strategy, or $\binom{2}{K}$ two-class subproblems by a one-against-one strategy. Then the work procedure of M^3-network can be launched, which consists of three steps: task decomposition, training base classifiers, and module combination.

Before the formal description of M^3-network, we will first briefly explain its mechanism (see Fig. 1). Suppose that Fig. 1(a) is the classification problem to be solved, in which small disks represent positive samples and small rectangles represent negative ones. Though simple at the first sight, it's actually a non-linear problem. To launch M^3-network learning, the samples of each class are first divided into two subsets, surrounded by dashed lines in the drawing. Then by pairing, four smaller classification subproblems shown in Figs. 1(b) through (e) are generated. One classifier is trained on each of the subproblems, and dashed lines represent the discriminant surfaces. Then with the $minimum$ operator, Fig. 1(f) is derived from Figs. 1(b) and (c), resulting in negative zone expanding; and in the same way, Fig. 1(g) is derived from Figs. 1(d) and (e). In the end, with the $maximum$ operator, Fig. 1(h) is derived from Figs. 1(f) and (g), resulting in positive zone expanding. In the outcomes of Fig.1(h), it can be found that a correct discriminant surface is generated.

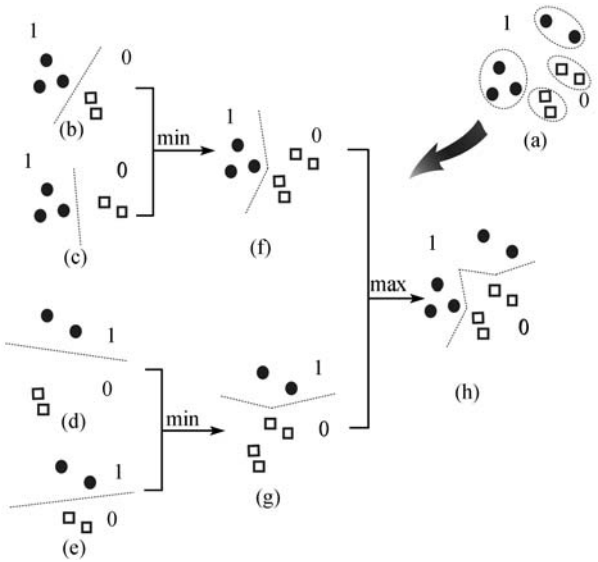

Fig. 1 Illustration of M^3-network

Now we will formally introduce the learning algorithm of M^3-network. Fig. 2 briefly illustrates the fine task decomposition and the corresponding module combination for a two-class problem. Here the learning of original two-class problem is first decomposed into $N_1 \times N_2$ subproblems. Then each subproblem yields a classifier, namely module, through the training of base classifier. Eventually an M^3-network, which can be used to classify new inputs, is generated by combining these modules. In this case, the M^3-network consists of $N_1 \times N_2$ individual modules, N_1 MIN

units, and one MAX unit. $\text{Min}^{i,*}$ shown in Fig. 2 expresses the MIN unit for the ith subset of the positive class. In the following three subsections, we will describe the procedure of training M^3-network.

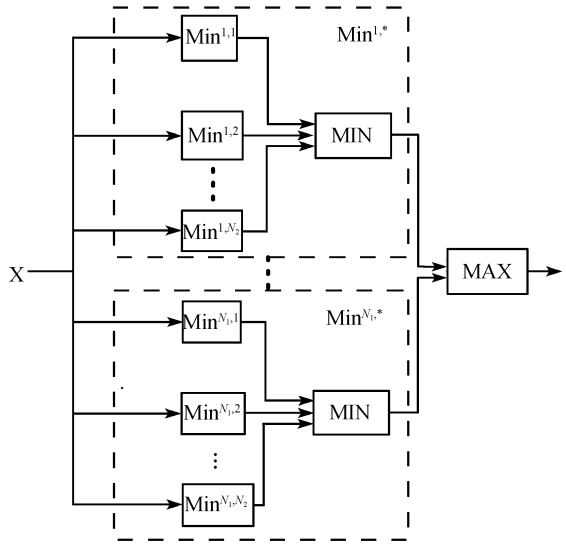

Fig. 2 Task decomposition and module composition of a two-class problem

2.1 Task decomposition

Let $\mathcal{T}_{ij}$ be the given training data set for a two-class classification problem,

$$\mathcal{T}_{ij} = \{(X_l^i,\ +1)\}_{l=1}^{L_i} \cup \{(X_l^j,\ -1)\}_{l=1}^{L_j},$$
$$\text{for } i = 1, 2,\ \ldots,\ K \text{ and } j = i+1, i+2,\ \ldots,\ K \quad (1)$$

where $X_l^i \in \mathcal{X}_i$ and $X_l^j \in \mathcal{X}_j$ are the training inputs belonging to class $\mathcal{C}_i$ and class $\mathcal{C}_j$, respectively; $\mathcal{X}_i$ is the set of training inputs belonging to class $\mathcal{C}_i$; L_i denotes the number of data in $\mathcal{X}_i$; $\bigcup_{i=1}^{K} \mathcal{X}_i = \mathcal{X}$; and $\sum_{i=1}^{K} L_i = L$. In this paper, the training data in a two-class problem are called *positive* training data if their desired outputs are $+1$. Otherwise, they are called *negative* training data.

Although the two-class problems defined by Eq. (1) are smaller than the original K-class problem, yet this partition may be not adequate for parallel learning. Since the number of samples in each class may vary largely, the sizes of these two-class problems can be quite different. Thus the training period can be delayed by some larger problems. Moreover, a large class and a small class will form an imbalanced classification problem – too many samples in one side – which will raise the difficult in training the classifiers [8]. To speed up training and improve classification accuracy, all the large and imbalanced two-class problems should be further divided into smaller and more balanced two-class problems.

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

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

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

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

$$\mathcal{T}_{ij}^{(u,v)} = \{(X_l^{(iu)},\ +1)\}_{l=1}^{L_i^{(u)}} \cup \{(X_l^{(jv)},\ -1)\}_{l=1}^{L_j^{(v)}},$$
$$\text{for } u = 1, 2,\ \ldots,\ N_i,\ v = 1, 2, \ldots,\ N_j,$$
$$i = 1, 2, \ldots,\ K, \text{ and } j = i+1, i+2 \ldots,\ K \quad (3)$$

where $X_l^{(iu)} \in \mathcal{X}_{iu}$ and $X_l^{(jv)} \in \mathcal{X}_{jv}$ are the training inputs belonging to class $\mathcal{C}_i$ and class $\mathcal{C}_j$, respectively;

$$\sum_{u=1}^{N_i} L_i^{(u)} = L_i \text{ and } \sum_{v=1}^{N_j} L_j^{(v)} = L_j.$$

2.2 Training base classifiers

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

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

$$\sum_{i=1}^{K-1} \sum_{j=i+1}^{K} N_i \times N_j, \quad (4)$$

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

$$\lceil L_i/N_i \rceil + \lceil L_j/N_j \rceil. \quad (5)$$

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 values of N_i and N_j. In this paper, traditional SVMs are selected as base classifiers and used to learn all the two-class subproblems.

2.3 Module combination

After all of the two-class subproblems defined by Eq. (3) have been learned by base classifiers, all the trained SVMs are integrated into an M^3-SVM with the MIN and MAX

units according to the following two combination principles: the minimization principle and the maximization principle [9].

• Minimization principle

Suppose a two-class problem $\mathcal{B}$ is divided into P smaller two-class subproblems, $\mathcal{B}_i$ for $i = 1, 2, \ldots, P$, and also suppose that all the two-class subproblems have the same positive training data and different negative training data. If the P two-class sub-problems are correctly learned by the corresponding P individual SVMs, M_i for $i = 1, 2, \ldots, P$, then the combination of the P trained SVMs with a MIN unit will produce the correct output for all the training inputs in $\mathcal{B}$, where the function of the MIN unit is to find a minimum value from its multiple inputs. The transfer function of the MIN unit is given by

$$q(x) = \min_{i=1}^{P} \mathrm{M}_i(x), \tag{6}$$

where x denotes the input variable.

• Maximization principle

Suppose a two-class problem $\mathcal{B}$ is divided into P smaller two-class sub-problems, $\mathcal{B}_i$ for $i = 1, 2, \ldots, P$, and also suppose that all the two-class subproblems have the same negative training data and different positive training data. If the P two-class subproblems are correctly learned by the corresponding P individual SVMs, M_i for $i = 1, 2, \ldots, P$, then the combination of the P trained SVMs with a MAX unit will produce the correct output for all the training input in $\mathcal{B}$, where the function of the MAX unit is to find a maximum value from its multiple inputs. The transfer function of the MAX unit is given by

$$q(x) = \max_{i=1}^{P} \mathrm{M}_i(x). \tag{7}$$

3 Patent classification

The classification task discussed in this paper is to classify Japanese patent documents on the section level of International Patent Classification (IPC). In this section, we will present the necessary background knowledge of patent classification, including the related work, the database we work on, and the IPC taxonomy.

3.1 Related work

Patents are playing more and more important roles in the progress of science and technology nowadays. As a result, automatic patent classification, which is a basic data mining technique on patents, has received much more attention. The European Patent Office tried a variety of preprocessing methods on patent data such as assigning different weights to patent sections and utilizing the citation between patents [10]. Larkey designed a system for searching and classifying U. S. patent documents based on query, and the K-nearest-neighbor algorithm was adopted in this system [11,12]. The Japan Patent Office developed a patent classification system based on keywords and used about 310000 patents in the training procedure. They achieved 96% classification accuracy in 38 categories and 82% classification accuracy in 2815 subcategories [13]. Fall and his colleagues compared the most commonly used classifiers on patent classification, such as naive Bayes, K-nearest-neighbor algorithm, support vector machine, neural networks, and decision rules, and reported that SVM provides the best performance [14,15].

3.2 Patent database

The corpus we work on here is provided by Japanese National Information Institutes' Testing Corpus for Information Retrieval (NTCIR): an organization which aims to evaluate the effectiveness of patent retrieval and classification from a scientific point of view. The corpus is publicly available for research purpose[†]. It includes Japanese patent applications over 10 years from 1993 to 2002 (approximately 3500000 documents), and each patent's document has a title and three fields: abstract, claim and description (see Table 1), among which the title and the claim are reported to be more valuable for classification.

3.3 IPC taxonomy

Once a patent application is submitted to a government patent office, it is usually assigned to one or more classification codes by human experts according to IPC. A patent has at least one IPC code, called main code, and maybe a set of secondary codes relating to other aspect expressed in the patent [16,17].

IPC is a complex hierarchical classification system, which divides all the technological fields into 8 sections, 120 classes, 630 subclasses and approximately 69000 groups. Fig. 3 illustrates the first four layers of IPC. The section layer is the top layer, and it has eight categories denoted by letters from A to H. The class layer is under the section layer and expressed as a section label followed by two digits. For example, 'A01'. The third layer is the subclass layer, which is represented as a class label followed by a capital letter. For example, 'A01B'. In general, current research has mainly concentrated on the top three layers since the definitions of layers below subclasses are still frequently changed.

[†] http://research.nii.ac.jp/ntcir/index-en.html

Table 1 The structure of Japanese patent documents

PATENT-JA-UPA-1998-000001	
⟨Bibliography⟩	
[publication date]	(43)【公開日】平成１０年（１９９８）１月６日
[title of invention]	(54)【発明の名称】土壌改良方法とその作業機
⟨Abstract⟩	
[purpose]	【課題】心土破砕、特に雪上心土破砕作業の際に積雪 ...
[solution]	【解決手段】心土破砕を行うために用いるサブソイラの ...
⟨Claims⟩	
[claim1]	【請求項１】 サブソイラ作業機を用いて心土破砕作業 ...
[claim2]	【請求項２】 サブソイラ作業機において、そのナイフ ...
⟨Description⟩	
[technique field]	【発明の属する技術分野】本発明は、土壌改良方法とそ ...
[prior art]	【従来の技術】圃場の表面がまだ積雪に覆われている状 ...
[problem to be solved]	【発明が解決しようとする課題】心土破砕は通常春先に ...
[means of solving problems]	【課題を解決するための手段】述のような目的達成す ...
[effects of invention]	【発明の効果】以上の説明から明らかなように、本発明 ...
...	...
⟨ Explanation of Drawing ⟩	
[figure1]	【図１】本発明を施す圃場断面図である。
...	...

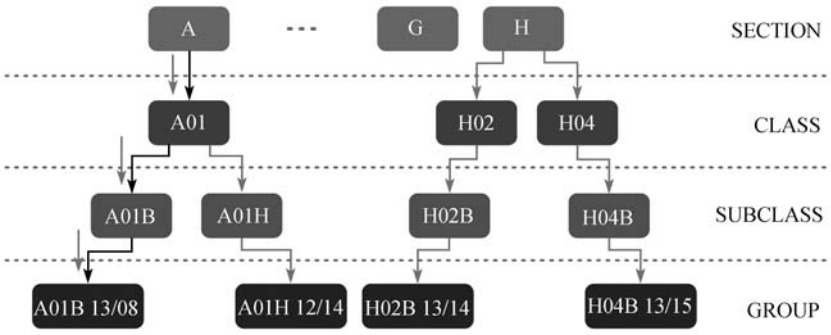

Fig. 3 Illustration of IPC taxonomy. Here, 'A' is the SECTION category label, 'A01' is the CLASS category label, 'A01B' is the SUBCLASS category label, and 'A01B 13/08' is the GROUP category label.

4 Decomposition of training data with prior knowledge

To justify our proposed method, let us first consider what an ideal division of a class's training data, so as to decompose the original classification task? Fig. 4 gives us a direct idea, that is, within the scope of a class, the similar samples should be together and put into the same subset. Therefore, the transferred problem is how to recognize the similarity among the samples?

In our previous work, we have proposed various division strategies. These methods fall into the following three categories. 1) random approach; 2) clustering based approach; and 3) explicit prior knowledge based approach. The random approach is a straightforward way and no any similarity concerning the training data is considered [7,9]. The clustering based approach is carried out in the feature space, and neighbouring is taken as similarity [18]. Explicit prior knowl-

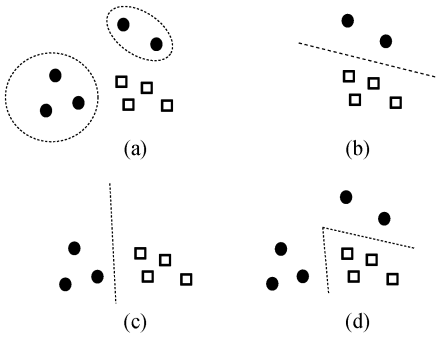

Fig. 4 Illustration of an efficient task decomposition way for M^3-network. (a) Original task, (b) sub-task I, (c) sub-task II, (d) solution of M^3- network

edge based apporoach make use of extra information abount the samples, which is usually acquired during collecting the samples. For example, when training a classifier to recognize the gender from human's facial images, we divided the training data by the human's age [19].

In this paper,we take the deep prior knowledge as clues, which makes the division much more reasonably. Our solution is that the samples with same attributes, which are derived from the prior knowledge, are similar ones and should be put together. This solution is kind of similar with the previous explicit prior knowledge based division approach, and the difference is the sorts of prior knowledge being used. In this paper, the deep knowledge of hieratical labels itself is utilized.

Fig. 4(a) shows the original classification problem, where small disks and small rectangles denote positive samples and negative samples, respectively. All the positive samples are divided into two subsets (surrounded by dotted lines), thus two subtasks are generated. In Figs. 4(b) and (c), two classifiers are trained on these two subtasks (dotted line for discriminant interface). Finally, in Fig. 4(d), M^3-network integrates the two classifiers and learns the original classification problem.

To simplify the description, we present our method in the context of patent classification. Note that our method can be easily applied to other databases with the substitution of the prior knowledge. Now suppose that we want to train an M^3-classifier to classify patent samples in the section level (see Section 3), the training data decomposition with publishing date and subclass label as prior knowledge, namely M^3-YC, can be described as the following three steps:

Step 1 Divide the data set of each section by published date, each subset for one year;

Step 2 Further divide the subsets by class, thus each subset for one class published in one year;

Step 3 Further divide the remained large subsets into subsets of fixed size randomly.

Figure 5 also illustrates this process. Different attributes among the samples are shown by colors and shapes. The colors are corresponding to publish dates, red samples being published in the same year and blue ones in another. The shapes are corresponding to the sub-categories, rectangles and triangles for two sub-categories of category A, and the circles and diamonds for two sub-categories of category B. The task decomposition consists of three steps, first by the prior knowledge of publish year (b), then by the sub-categories (c), at last by random for some remained large subsets (d). Typically, if the first two steps are removed from the above procedure, that is, the data set of each section is divided randomly at first, then it becomes the conventional M^3-SVM, namely M^3-Rand. In this paper, M^3-Rand to-

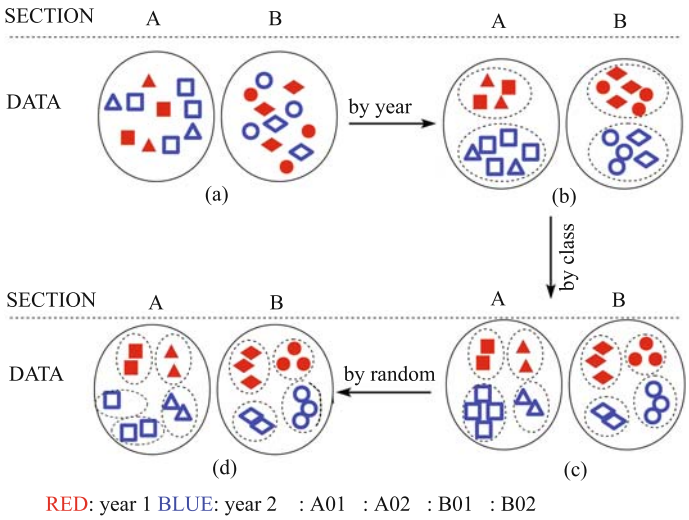

Fig. 5 Task decomposition with prior knowledge. (a) Original task, (b) first step decomposition result by publish year, (c) second step decomposition result by sub-categories, (d) third step decomposition result by random

gether with traditional SVMs serve as the baseline methods for comparing with our proposed method, M^3-YC.

Both M^3-Rand and M^3-YC have the process of random decomposition, through which the task of learning the original classification problem is eventually decomposed into sub-tasks with expected size. Normally a parameter, the target subset size, needs to be set to tune this process. If target size is too small, it will result in too many subproblems. If there are no enough number of CPUs available under this situation, the training and test time cost will be raised rather than being reduced. On the contrary, if the target size is too large, some subproblems will be too large and complicated for one base classifier to learn. According to our preliminary experiments, we eventually decided that the target size of subset is 2000 samples.

5 Experimental settings

5.1 Data sets

We had an extended version of NTCIR-5 Japanese patent database at hand, which consists of all the patent applications published by Japanese Patent Office from 1993 to 2002. So as to systematically evaluate three methods: SVMs, M^3-Rand, and M^3-YC, we made eight pairs of data sets (see Table 2). The test sets were fixed as we wanted all the results to be comparable. The training set ended just before the test set as the most recent patent records were the most useful. The start points of training sets varied so the sizes of training sets changed.

Table 2 Description of training and test data sets

number of years	years		set size	
for training	training	test	training	test
1	2000	2001,2002	358072	733570
2	1999,2000	2001,2002	714004	733570
3	1998–2000	2001,2002	1055391	733570
4	1997–2000	2001,2002	1386850	733570
5	1996–2000	2001,2002	1727356	733570
6	1995–2000	2001,2002	2064325	733570
7	1994–2000	2001,2002	2415236	733570
8	1993–2000	2001,2002	2762563	733570

5.2 Feature extraction and filtering

Several steps of preprocessing need to be done so as to get the vector representation of the Japanese text. They are tokenization, term filtering, and term indexing.

† http://chasen.naist.jp/hiki/ChaSen/

• Tokenization

This step is to generate a list of clean and informative words from each patent document. Four fields of raw patent text – Title, Abstract, Claim and Description – are extracted from each structured patent document (see Table 1), as they are the parts most informative about the patent's content. Then these texts are segmented into isolated words by using the software Chasen†. After that, the stop words need to be removed from the results. The remained words, namely *term*s in the research domain of text categorization, are the features for successor classification task. Table 3 shows the result we get from the example illustrated in Table 1.

Table 3 The structure of Japanese patent documents

土壌 (soil) 改良 (improve) 方法 (methods) 作業 (working)
機 (machine) % title
心土 (subsoil) 破砕 (crack) 雪上 (snow) 心土 破砕
作業 ··· % abstract
ｻﾌﾞｿｲﾗ (subsoiler) 作業 機 心土 破砕 作業 ··· %claim
発明 (patent) 土壌 改良 方法 ··· %description

• Term filtering

This step is to remove the useless terms for classification task. Not all the terms occurring in a document can provide information on the document's class. With these terms removed, the number of the representation vectors' dimensions is reduced, thus the computational cost will be cut down and the generalization error will be reduced. There are three popular judgment criteria of terms in the TC domains– χ_{avg}, χ_{max} and Information Gain (IG) [20].

The χ^2 is a statistic measurement of the lack of independence between two random variables, the term t and the class c. It can be compared to the χ^2 distribution with one degree of freedom to judge extremeness. The formula of χ^2 for a binary classification problem is:

$$\chi^2(t,c) = \frac{\|T_r\|(n_{tc}n_{\bar{t}\bar{c}} - n_{t\bar{c}}n_{\bar{t}c})^2}{n_t n_{\bar{t}} n_c n_{\bar{c}}}, \tag{8}$$

where T_r is the training corpus, n_{tc} is the number of samples that belong to c and have t, $n_{\bar{t}c}$ is the number of samples that belong to c but don't have t, and the rest may be deduced by analogy. As for the multiclass problem, there are two extensions based on the mechanism to integrate the measurement on each class:

$$\chi^2_{max}(t) = \max_{c\in C} \chi^2(t,c), \tag{9}$$

$$\chi^2_{avg}(t) = \underset{c\in C}{avg}\, \chi^2(t,c), \tag{10}$$

where C is the whole set of classes.

IG measures the decreased bits of the information of a document's classes by knowing the presence or absence of certain term in this document.

$$IG(t) = Info(c) - Info(c|t) = -\sum_{c\in C} P_r(c)logP_r(c) + P_r(t)\sum_{c\in C} P_r(c|t)log P_r(c|t) + P_r(\bar{t})\sum_{c\in C} P_r(c|\bar{t})logP_r(c|\bar{t}), \quad (11)$$

where $Info(c)$ is the bits of information of c, $Info(c|t)$ is the bits of conditional information of c knowing t, and $P_r(x)$ is the possibility of x occurrence.

We first decide which criterion to adopt by the pilot experiments, with certain number of top terms under one criterion being used for training and test. The χ^2_{avg} approach turned out to be the best. Then we decide the number of features to use by trying the figures 2500, 5000, 7500, ..., 160000, and 5000 is found to be the smallest number which could give nearly top performance. Table 4 shows the top 10 terms sorted by χ^2_{avg}, most of which are technique terms as patent documents mainly deal with technological stuffs.

Table 4 Top 10 terms selected by χ^2_{avg}

terms	explanation	χ^2_{avg}
データ	data	10384.72
情報	information	10199.42
回路	circuit	9561.67
信号	signal	8387.75
記録	record	7901.17
物	article	7528.72
含有	contain	7374.12
接続	connect	7324.43
絶縁	insulation	7194.85
基板	baseplate	7076.72

• Term indexing

This step is to generate the real numerical vectors. Suppose there are n unique terms occurring in the training corpus, then the vectors of the samples will be n in length, with each dimension corresponding to one term. As to compute the weight – extent of importance – of a term to a document, there exist several methods and we adopt the dominant one – TFIDF [21]:

$$tfidf(t,d) = n(t,d)\log\frac{|T_r|}{n_{T_r}(t)}, \quad (12)$$

where t denotes a term, d denotes a document, T_r denotes the training corpus, $n(t,d)$ denotes the number of times t occurs in d, namely term frequency, and $n_{Tr}(t)$ denotes the number of documents where t occurs, namely document frequency. TFIDF is actually in the style of Information Retrieval, which embodies two intuitions that (i) the more often a term occurs in a document, the more it is representative of its content, and (ii) the more documents a term occurs in, the less discriminating it is. Table 5 shows the vector representations of some patent documents.

Table 5 The vector representations of patent documents. The format of The vectors adopted by SVM^{light} is taken, that is, *vetor:=(dimension:value)+*

No.	Vectors				
1	72 : 0.730	98 : 1.790	138 : 1.310	141 : 4.495	...
2	28 : 26.353	29 : 9.232	31 : 2.795	71 : 1.463	...
3	71 : 1.463	79 : 2.441	85 : 2.993	113 : 11.393	...
4	42 : 2.164	60 : 0.905	109 : 2.061	138 : 2.947	...
5	28 : 7.529	72 : 6.577	139 : 8.103	167 : 8.728	...

5.3 Computational platform

All of the experiments were performed on a Lenovo cluster system consisting of three fat nodes and thirty thin nodes. Each fat node has 32 GB of RAM and two 3.2-GHz quad-core CPUs, while each thin node has 8 GB of RAM and two 2.0-GHz quad-core CPUs. Experiments with the conventional SVMs were performed on the fat nodes, because they need large memory, while experiments with the M^3-SVMs were done on the thin nodes, because each sub-problem was small and a lot of processors were required for parallel training.

5.4 Training base classifier

M^3-network is just a general framework for pattern classification, and some classifiers, namely base classifiers, need to be embedded into it. There were several alternative choices, such as artificial neural networks [6,7], naive Bayes, K-nearest-neighbor algorithm [22,23], and SVMs. Here conventional SVMs were adopted for its well-known excellent performance on classification tasks as well as text categorization. For the trade-off of performance and cost, the linear kernel was taken. The kernels of Radius Base Function and Polynomial would hopefully provided better performance, but they would cost overwhelmingly long time in training conventional SVMs. The SVM^{light}, an implement of SVM by Joachims [24], was adopted for it has been specially optimized on text categorization tasks. It actually played two roles in this paper, the baseline method and the base classi-

fiers for M^3-SVMs.

5.4 Performance measurement

The convention performance measurement of accuracy in machine learning domain turns out to be less rational for most TC tasks. Instead, the function F_1 is usually used in TC domain. The formula of F_1 of one class is [21]:

$$F_1 = \frac{2PR}{P+R}, \tag{13}$$

$$P = \frac{TP}{TP+FP}, \tag{14}$$

$$R = \frac{TP}{TP+FN}, \tag{15}$$

where TP is the number of classifier's true positive predictions, FP is the number of false positive predictions, and FN is the number of false negative predictions. P is named the precision, and R is named the recall. In practise, there are the two following versions of F_1, depending on the strategy of integrating all the results of individual samples:

$$micro - F_1 = \frac{2PR}{P+R}, \tag{16}$$

$$macro - F_1 = \underset{c\in C}{avg} \frac{2P_cR_c}{P_c+R_c}, \tag{17}$$

where P and R are computed with all the classes, while P_c and R_c are computed on only classes c. For example, sample s actually belongs to classes c_1 and c_2, while the classifier's predictions is c_2 and c_3, then in computing $micro-F_1$, TP, FP and FN will be increased by 1, while in computing P_{c1} and R_{c1} for $macro - F_1$, only FN will be increased by 1.

6 Results and discussion

6.1 Accuracy

Figure 6 presents the experimental results, with $micro - F_1$ and $macro - F_1$ as accuracy measurement.

The following conclusions can be drawn from these results:

1) On the aspect of test accuracy, the two M^3-SVM methods, M^3-Rand and M^3-YC, are both superior to conventional SVMs. We can learn from the training scores that SVMs with linear kernel are unable to learn the training set completely, because its $micro - F1$ and $macro - F1$ are only about 80%. On the contrary, as an ensemble learning algorithm, M^3-SVMs can generate powerful classifier by combining simple classifiers. As a result, M^3-SVMs have fulfilled the learning on all the training sets with the accuracies of nearly 100%.

2) M^3-Rand and M^3-YC show superior robustness to conventional SVMs over dated samples. Along the moving backward of the starting time point of training set, more and more dated samples are added, thus the performance of conventional SVMs decreases. Contrarily, the performance of two M^3-SVMs algorithms become better and better at the same time. Though their performance deceases a little in the 5th year point and 7th year point, yet the ranges are very small and have little impact on the curves' overall trend.

3) M^3-YC over-performs M^3-rand on each year point, which indicates incorporating prior knowledge of publishing date and subclass into task decomposition will undoubtedly improve the classification performance.

Something needs to be addressed here. The difference of M3-YC and M3-Rand in performance is small, which is kind of disappointing. The reason may be that the label information of training samples in subclass is not adequate for division. In fact, subclasses of IPC are extremely imbalanced, and many subsets are larger than predefined set size, in which randomly dividing has to be performed. For example, in the training set of eight years, class G contains 216105 samples and 13 subclasses. However, the two largest subclass of G06 (62767 samples, 29%) and G01 (48700 samples, 23%) own more than a half of the overall samples, which have no choice but to be divided randomly. Thus, M3-YC and M3-Rand will lead to similar training subsets, which draw their performance close. Anyhow, from the experimental results here, incorporating prior knowledge into division does undoubtedly improve the classification accuracy. IPC is a hierarchical structure with six levels, and with the information of these deep subclasses, the training set can be adequately divided, thus less random division will be needed. We are to carry out these experiments soon, and more improvement can be expected.

In addition, the data sets that our method applies to are required to have hierarchical structures (temporal information may also be used if existing). Nowadays, many data sets other than patent documents are organized in such structures. For example, the new version of Reuters corpus – RCV1– employs a hierarchical structure and the detailed description can be found in its release page[†] [25]. Another example is web pages where a uniform hierarchical structure of ODP is usually adopted (Open Directory Program[‡]) [26].

[†] http://jmlr.csail.mit.edu/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm

[‡] http://www.dmoz.org/about.html

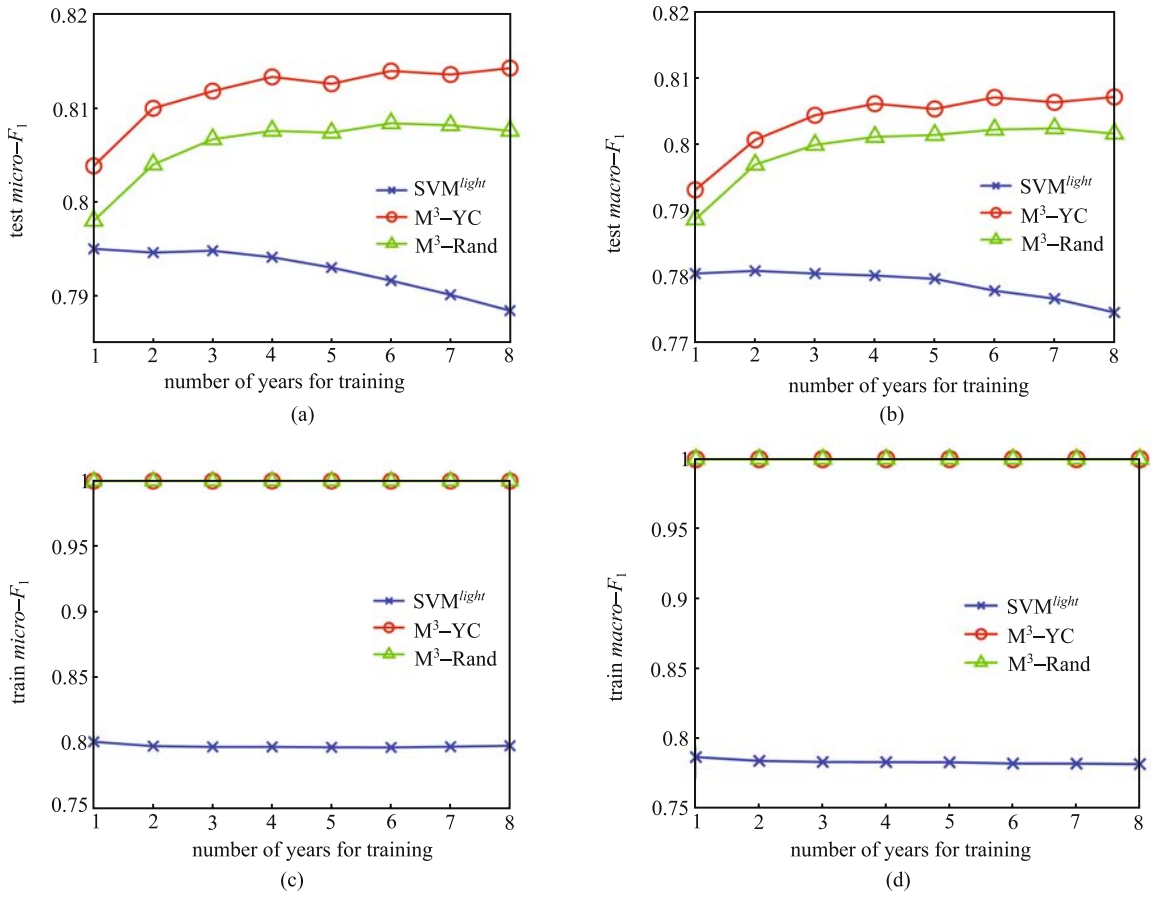

Fig. 6 Performance comparison of SVMs, M^3-Rand and M^3-YC: (a) $micro-F_1$ on test data; (b) $macro-F_1$ on test data; (c) $micro-F_1$ on training data; and (d) $macro-F_1$ on training data

6.2 Time cost

As mentioned in Section 1, M^3-SVMs (both M^3-Rand and M^3-YC) have the merit of parallel computing, which can speed up the learning of classification problems. However, the Lenovo parallel computer was actually being shared by other users while we were conducting the experiments, so the accurate time cost of M^3-SVMs could not be measured, while the time cost of SVMs was recorded since it didn't involved parallel running.

We evaluate the time cost of M^3-YC by using the following formula:

$$t_{M^3} = \frac{n_{mod} \times t_0}{n_{cpu}}, \tag{18}$$

where n_{mod} is the number of modules (or subproblems) generated by M^3-SVMs, t_0 is the average time cost of per module, and n_{cpu} is the number of CPUs. During our experiments, n_{mod} have been recorded automatically and n_{cpu} is 30 on Lenovo computer. As for t_0, it must be measured by experiment, and the eventual value we get is 0.025 second per module.

Figure 7 presents the time costs of SVMs and M^3-YC in our experiments. From these results, we can find out that the M^3-YC's time cost is only about 1/10 of the SVM's, which agrees with our expectation well.

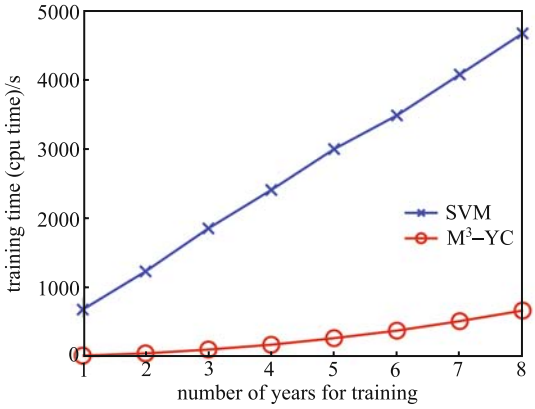

Fig. 7 Time cost of SVMs and M^3-YC

7 Analysis on fault samples

In this section, we analyze the experimental results mentioned above by observing and comparing the outputs of three methods on certain samples. Our object is to find out M^3-YC's advantages and disadvantages, so as to seek any possibility to further improve it. In the first half of this section we work on the contingency tables, and in the second half we go deep into analysis of individual patent samples.

7.1 Contingency table

In statistics, contingency tables are used to record and analyze the relationship between two or more variables, most usually categorical variables. They can be taken as useful tools in analyzing the results of classification experiments [21]. The contingency tables of three different methods are presented in Tables 6, 7, and 8. The values in the diagonal line represent the proportion† of samples correctly categorized, while the others represent the proportions of the samples wrongly categorized by the way corresponding to their positions in the table. From these tables, the following two observations can be obtained.

1) From the viewpoint of data set, we can find out which classes of samples are easy to be recognized and which classes of samples are confusing, specially, with which classes they are likely to mix.

2) From the viewpoint of classification methods, we can tell on which classes a classification method is strong, and on which classes it is weak, further, for which classes it might mistake these samples.

Table 6 SVM's contingency table

	A	B	C	D	E	F	G	H
A	743	34	22	5	15	14	33	6
B	101	1389	69[a)]	16	51	81	92	49
C	38	94	734	11	7	16	33	24
D	8	8	3	83[b)]	1	1	1	1
E	23	27	8	1	349	22	14	6
F	23	71	14	2	21	645	18	20
G	54	124	32	2	15	38	2056	234
H	25	85	61	2	11	40	289	1884

Note:
a) The samples involved by B-th row and C-th column are the samples of class B and wrongly categorized into class C. The ratio of such samples over the whole data set is 0.0069;
b) The samples involved by D-th row and D-th column are the samples of class D being correctly categorized.

† The number is multiplied by 10^4 for pretty printing, e.g. 15 means the ratio of 0.0015

We would follow these two viewpoints respectively in the discussions below.

Table 7 M^3-Rand's contingency table

	A	B	C	D	E	F	G	H
A	797	50	22	6	17	15	38	8
B	75	1450	85	11	40	86	110	66
C	29	68	754	5	5	12	30	34
D	8	11	5	97	1	2	2	1
E	20	28	6	1	373	18	14	7
F	20	59	12	1	18	671	20	26
G	47	107	23	1	10	28	2092	240
H	18	59	38	1	6	26	232	1843

Table 8 M^3-YC's contingency table

	A	B	C	D	E	F	G	H
A	789	45	20	5	16	14	33	7
B	79	1461	83	11	40	83	98	60
C	29	64	757	5	5	11	27	32
D	8	10	5	97	1	2	2	1
E	20	26	6	1	372	17	12	6
F	21	57	11	1	18	674	19	23
G	51	108	24	2	10	28	2126	228
H	18	60	38	1	6	27	220	1868

• Viewpoint of data sets

Though these contingency tables, Tables 6, 7 and 8, are computed from the results of different classification methods, yet their similarity are obvious. Such similarity, which lies under various classification systems, can be considered as the inside characteristics of data set. So as to simplify the analysis, we come up with a new table, Table 9 by averaging the values of the three previous tables.

A discovery from Table 9 is that the values of (G,H) and (H,G) are the largest among all the fault cases, which indicates these two classes are easy to mixed up. According to the authority documents of IPC, class G is Physics Instruments, and class H is Electricity. Since both of these two are foundation of modern technology, it is understandable that there exist lots of intersections between them. For example, G08, a subclass of G, was signaling which contains collecting, transmission, and display of signals, while electronics is undoubtedly the main technique to fulfill these tasks. On the contrary, H03, a subclass of H, is basic electronic circuitry which contains making electronic component, designing amplifying circuits, and these subjects are all based on physical techniques.

Table 9 The averaged contingency Table of SVMs, M^3-Rand, and M^3-YC

	A	B	C	D	E	F	G	H
A	777	43	21	5	16	14	35	7
B	85	1433	79	13	44	84	100	58
C	32	75	748	7	6	13	30	30
D	8	10	4	93	1	2	1	1
E	21	27	7	1	364	19	13	6
F	21	63	13	1	19	663	19	23
G	51	113	26	2	12	31	2091	234
H	20	68	46	1	8	31	247	1865

• Viewpoint of classification methods

To simplify the comparison between classification methods, differential contingency tables are made (see Tables 10 and 11).

Table 10 The differential contingency table of M^3-YC and SVM

	A	B	C	D	E	F	G	H
A	+46	+11	−2	0	+2	0	0	+1
B	−22	+72	+15[a)]	−5	−11	+3	+6	+10
C	−9	−29	+22	−6	−1	−4	−6	+7
D	0	+2	+2	+14	0	+1	+1	0
E	−2	−1	−2	0	+23	−5	−1	0
F	−2	−14	−3	−1	−3	+28	0	+3
G	−3	−16	−8	0	−5	−10	+70	−5
H	−7	−25	−23	−1	−5	−413	−69	−16

Note: a) e.g., This figure meant the ratio of classifying class B's samples into class C is increased by 0.0015 in M^3-YC compared to SVM

From the differential table between M^3-YC and SVM, it can be found that the values in the diagonal line are all positive, which indicates M^3-YC raise accuracy on all the classes. Moreover, the value of (A,B) is positive and that of (B,A) is negative, which indicates M^3-YC classified more class A's samples into class B and less class B's samples into class A, which can be interpreted as it raises the recall of class B and lowers that of class A.

Similar analysis can be performed on the differential tables of M^3-YC and M^3-Rand. M^3-YC shows a rise in the classes of B, C, E, F, G and H, while it shows decent in class A. Besides, more samples of class B and F are wrongly classified into class A by M^3-YC. It should be noted that the values of both (G,H) and (H,G) are negative, which indicates that M^3-YC can handle the samples of classes F and H much better. This improvement plays an key role in raising the global classification accuracy.

Table 11 The differential contingency table of M^3-YC and M^3-RAND

	A	B	C	D	E	F	G	H
A	−8	−5	−2	0	−1	−1	−5	−1
B	+4	+11	−1[a)]	0	+1	−3	−12	−6
C	0	−4	+3	0	0	0	−3	−2
D	0	−1	0	0	0	0	0	0
E	0	−1	0	0	−1	−1	−1	−1
F	+1	−2	−1	0	0	+3	−2	−3
G	+4	+1	+1	0	+1	0	+35	−12
H	0	+1	0	0	0	+2	-12	+25

Note: a) e.g., This figure meant the ratio of wrongly classifying class B's samples into class C is decreased by 0.0001 in M^3-YC compared to M^3-Rand

7.2 Individual patent documents

We have already known some global characteristics of patent documents, such as which two classes are easy to be mixed up, and on which classes the classification methods tend to make mistakes. In this section we will directly read some patent documents, so as to find the cause of these phenomena. After some analysis, we categorize the puzzle patent documents into two sorts.

The first sort is the kind of patent documents which are of great possibility of being wrongly classified. In most cases they contain typical words of irrelevant classes, so any classification method with terms as features tends to make wrong predictions on them. On the contrary, there exist some other patent documents which were only partly similar to some irrelevant classes. As they lie in the margin zone of being classified wrongly and being classified correctly, we named them marginal patent documents. The patent documents of the first sort are wrongly classified by all the classification methods, while the patent documents of the second sort are correctly classified by some methods and wrongly classified by the others.

• Mistakable patent documents

Class G and class H are found to be easily mixed up in previous analysis on contingency tables (Section 7.1). So we pick out two patent documents of class H, which were wrongly classified into class G by all the three methods, SVMs, M^3-Rand and M^3-YC (Tables 12 and 13)

Example I is certainly very confusing, for the physical technique term of harden (硬化), insulation (绝缘) and intensity (强度) frequently appear in this patent document, so it is not strange at all that it is mistaken as class G. Example II belongs to both class G and class H, which can be detected from its IPC labels. As a result, it is acceptable that

Table 12 Example I: class H's sample wrongly classified into class G

ID	PATENT-JA-UPA-2001-006462
Title	重合 皮膜 形成 方法 用い 金属 絶縁 被覆 方法 及び 絶縁 被覆金属 導体
IPC	H01B/13/16 C08F/2/58 ...
Abstract	従来 電解 重合 法 用い 絶縁 性 高分子 薄膜 形成 皮膜 強度基 材 密着 性 等 物性向上 後処理 行っ 事例 存在 官能 性 残 ...
Claim	官能 性 残 基 有する ポリマ 電解 重合 形成 電解 中 もしくは 電解 後 前記 ポリマ 硬化 剤 添加 その後 前記 ポリマ 硬化 行う 特徴 重合 ...
Description	本 発明 金属 コイル 等 表面 施す 電気 絶縁 性 皮膜 形成 方法 特に 電解 重合 法 有機 高分子 皮膜 形成 方法用い 金属 材料 絶縁 被覆 方法 絶縁 被覆 金属 導体 近年電子 ...

Table 13 Example II: class H's sample wrongly classified into class G

ID	PATENT-JA-UPA-2001-006485
Title	自動販売 機 用 押釦スイッチ (Switch)
IPC	H01H/13/14 G07F/9/00 ...
Abstract	自動 販売 機 リニユアル 時 回路 基板 水密 構造 維持 ケース レンズ 交換 いたずら ケース 内部 部品 破損 防止 リニユアルコスト 低減 可能 自動 販売 機 用 押釦 スイッチ 提供 自動 ...
Claim	自動 販売 機 前面 パネル 表側 取り付け 自動 販売 機 用押釦 スイッチレンズ 孔 有し ケース ベース 組み合わせる両者 間 ...
Description	本 発明 自動 販売 機 前面 パネル 表側 取り付け 自動 販売 機 用 押釦 スイッチ 特に 自動 販売 機 リニユアル 時 回路基板 水密 構造 維持 ケース レンズ 交換 ...

this patent document is classified into class G, which is considered to be a correct prediction in the context of multi-label classification.

• Marginal patent documents

Here we also give two examples, both of which belong to class H and were mistaken for class G by SVMs or M^3-Rand, while correctly classified by M^3-YC (see Tables 14 and 15). Example III contains many common words which have moderate tendency towards some other classes, such as mechanism (機能), discrimination (区分) and model (モード), and those words together make it become a marginal patent document. Example IV's topic electron tub (陰極線管) happens to be a typical word for both class G (physical instruments) and class H (electronics), which is the reason why it became a marginal patent document of both class G and class H.

Table 14 Example III: class H's sample mistaken for class G only by SVM

ID	PATENT-JA-UPA-2001-006480
Title	多 機能 スイッチ (switch 开关) 装置
IPC	H01H/13/02 B60R/16/02 ...
Abstract	車両 各 部位 装備 中 操作 特定 部位 装備 容易 発見 各種 スイッチ 機能 中 特定 部位 装備 対応 所望 スイッチ 機能簡単確実 選出 操作 多 機能 スイッチ 装置 提供 ...
Claim	所定 選択 操作 応じ 各種 スイッチ モード 何れ 択一 的 選択 モード 選択スイッチ モード 選択 スイッチ選択 スイッチ モード 対応 複数 区分 選択 画面 成る ...
Description	本 発明 例えば 車両 各種 装備 動作 多 機能 スイッチ 装置 係り 特に 車両 各 部位 装備 中 操作 特定 部位 装備 容易 発見 各種 スイッチ 機能 中 特定 部位 ...

Table 15 Example IV: class H's sample mistaken for class G by SVM and M^3-Rand

ID	PATENT-JA-UPA-2001-006544
Title	カラー 陰極線管 製造 用いる 露光 装置
IPC	H01J/9/227 H01J/31/00
Abstract	発明 分割 走査 方式 カラー 陰極線管 適用 複数 分割 領域 境界 ズレ ない 連続 良質 画像 出力 蛍光 面 露光 装置 提供 課題 電子 銃 備え 分割 方式 カラー ...
Claim	蛍光 面 複数 分割 領域 分割 走査 各 分割 領域 描か 複数 カラー 画像 合成 1 つ カラー 画像 表示 分割 走査 方式 カラー 陰極線管 製造 用い 露光 装置 上記 ...
Description	発明 カラー 陰極線管 パネル 内面 蛍光 面 形成 露光 装置 係り 特に 蛍光 面 複数 分割 領域 分割 走査 各 分割 領域 描か カラー 画像 合成 1 つ カラー 画像 ...

8 Conclusions

In this paper, we have investigated the idea of incorporating prior knowledge into learning by dividing the training data. We have proposed a new task decomposition method for M^3-SVMs. To testify our proposed method, we apply it to the task of patent classification with samples' publishing date and labels' hierarchical structure as prior knowledge, namely M^3-YC. Two other methods, conventional SVMs and conventional M^3-SVM with random task decomposition strategy, namely M^3-Rand, are taken as baseline methods. The experimental results show that M^3-YC always over-performs SVMs and M^3-Rand, which demonstrate incorporating prior knowledge into learning can efficiently raise the classification accuracy.

Moreover, the research with the new method on patent

classification brings us the following two conclusions, which may be useful for further real-world applications.

1) The problem of patent classification is not linear separable, since conventional SVMs with linear kernel can only achieve training accuracies about 80%, while both M^3-Rand and M^3-YC reach training accuracy of nearly 100%. As a result, the M^3-SVMs methods are superior to conventional SVMs in generalization accuracy.

2) The dated patent samples actually contain noises, which can harm the classification performance. It may be caused by the revise of IPC standard, evolvement of the language style, or the shift of research focus. Both M^3-Rand and M^3-YC methods are robust to such noises while conventional SVMs are not. Along with more and more samples added into the training set, the classification accuracy of conventional SVMs decreases, while those of M^3-Rand and M^3-YC increase instead.

Acknowledgements This research was partially supported by the National Natural Science Foundation of China (Grant No. 60773090) and the Fujitsu Research and Development Center Co., Ltd., Beijing, China.

References

1. Liu B, Li X L, Lee W S, Yu P S. Text classification by labeling words. AAAI, 2004
2. Wu X Y, Srihari R. Incorporating prior knowledge with weighted margin support vector machines. In: Proceedings of International Conference on Knowledge Discovery and Data Mining, 2004, 326–333
3. Schapire R E, Rochery M, Rabim M, Gupta N. Boosting with prior knowledge for call classification. IEEE Transactions on Speech and Audio Processing, 2005, 13, 174–181
4. Zhu J B, Chen W L. Improving text categorization using domain knowledge In: Proceedings of International Conference on Applications of Natural Language to Information Systems, 2005, 103–113
5. Dayanik A, Lewis D D, Madigan D, Menkov V, Genkin A. Constructing informative prior distributions from domain knowledge in text classification. In: Proceedings of ACM'S Special Interest Group on Information Retrieval, 2006
6. Lu B L, Ito M. Task decomposition based on class relations: a modular neural network architecture for pattern classification. Biological and Artificial Computation: From Neuroscience to Technology. Springer, LNCS, 1997, 1240: 330–339
7. 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, 1999, 10: 1244–1256
8. Anand R, Mehrotra K G, Mohan C K, Ranka S. An improved algorithm for neural network classification of imbalanced training sets. IEEE Transaction on Neural Netwook, 1993, 4: 962–969
9. Lu B L, Wang K A, Utiyama M, Isahara H. A part-versus-part method for massively parallel training of support vector machines. In: Proceedings of International Joint Conference on Neural Networks, 2004, 735–740
10. Krier M, Zaccá F. Automatic categorization applications at the European patent office. World Patent Information. Elsevier, 2002, 24(3): 187–196
11. Larkey L. Some issues in the automatic classification of US patents. Learning for Text Categorization. Technical Report WS-98-05, 1998, 87–90
12. Larkey L. A patent search and classification system. In: Proceedings of the fourth ACM conference on Digital libraries, 1999, 179–187
13. Mase H, Tsuji H, Kinukawa H, Ishihara M. Automatic patents categorization and its evaluation. Transactions of Information Processing Society of Japan(IPSJ), 1998
14. Fall C J, Benzineb K. Literature survey: Issues to be considered in the automatic classification of patents. World Intellectual Property Organization, 2002, 29
15. Fall C J, Torcsvári A, Benzineb K, Karetka G. Automated categorization in the international patent classification. In: Proceedings of ACM'S Special Interest Group on Information Retrieval. New York: ACM Press, 2003, 37: 10–25
16. Fujii A, Iwayama M, Kando N. Test collections for patent retrieval and patent classification in the 5th NTCIR workshop. In: Proceedings of the 5th international conference on language resources and evaluation, 2004, 1643–1646
17. Fujii A, Iwayama M, Kando N. Introduction to the special issue on patent processing. Information Processing and Management, 2007, 1149–1153
18. Wen Y M, Lu B L, Zhao H. Equal clustering makes min-max modular support vector machine more efficient. In: Proceedings of International Conference on Neural Information Processing, 2005, 77–82
19. Lian H C, Lu B L, Takikawa E, Hosoi S. Gender recognition using a min-max modular support vector machine. In: Proceedings of International Conference on Natural Computation, 2005, 438–441
20. Yang Y M, Pedersen J O. A comparattive study on feature selection in text categorization. In: Proceedings of International Conference on Machine Learning, 1997, 187–196
21. Sebastiani F. Machine learning in automated text categorization. ACM Computing Surveys, 2002, 34: 1–47
22. Zhao H, Lu B L. A modular k-nearest neighbor classification method for massively parallel text categorization. In: Proceedings of First International Symposium on Computational and Information Science. Springer, LNCS, 2004, 3314: 867–872
23. Wu K, Lu B L, Uchiyama M, Isahara H. An empirical comparison of min-max-modular k-NN with different voting methods to large-scale text categorization. Soft Computing − A Fusion of Foundations, Methodologies and Applications, 2008, 12(7): 647–655
24. Joachims T. Making large-scale support vector machine learning practical. Advances in Kernel Methods: Support Vector Learning. Cambridge: MIT Press, 1998
25. Lewis D D, Yang Y, Rose T, Li F. RCV1: A new benchmark collection for text categorization research. Journal of Machine Learning Research, 2004, 5: 361–397
26. Liu W, Xue G R, Yu Y, Zeng H J. Importance-based web page classification using cost-sensitive SVM. In: Proceedings of International Conference on Web-Age Information Management, 2005, 127–137