Single-shot Detection of Multiple Categories of Text using Parametric Mixture Models

Naonori Ueda
NTT Communication Science Laboratories
2-4 Hikaridai, Seika-cho, Soraku-gun,
Kyoto Japan
ueda@cslab.kecl.ntt.co.jp

Kazumi Saito
NTT Communication Science Laboratories
2-4 Hikaridai, Seika-cho, Soraku-gun,
Kyoto Japan
saito@cslab.kecl.ntt.co.jp

ABSTRACT

In this paper, we address the problem of detecting *multiple* topics or categories of text where each text is *not* assumed to belong to one of a number of mutually *exclusive* categories. Conventionally, the binary classification approach has been employed, in which whether or not text belongs to a category is judged by the binary classifier for every category. In this paper, we propose a more sophisticated approach to *simultaneously* detect multiple categories of text using *parametric mixture models (PMMs)*, newly presented in this paper. PMMs are probabilistic generative models for text that has multiple categories. Our PMMs are essentially different from the conventional mixture of multinomial distributions in the sense that in the former several basis multinomial parameters are mixed in the *parameter space*, while in the latter several multinomial components are mixed. We derive efficient learning algorithms for PMMs within the framework of the maximum a posteriori estimate. We also empirically show that our method can outperform the conventional binary approach when applied to multi-topic detection of World Wide Web pages, focusing on those from the "yahoo.com" domain.

1. INTRODUCTION

As a large quantity of text is being stored in the World Wide Web, electric mail, digital libraries, and so on, automatic text categorization is becoming a more important and fundamental task in information retrieval and text mining. In particular, since a document usually consists of several topics, detecting multi-topic or multi-category of text is of great practical value. Hence, this type of detection problem has become a challenging research theme in the field of machine learning.

This detection problem is different from the traditional pattern classification problems such as character recognition and speech recognition in the sense that each sample is *not* assumed to be classified into one of a number of predefined *exclusive* categories. For the multi-category detection problem, conventionally, a binary classification approach has been utilized along with state-of-the-art methods such as support vector machines (SVMs) [15][6] and naive Bayes (NB) classifiers [8][10]. In this approach, the multi-category detection problem is decomposed into separate binary classification problems. It has been reported that SVM can provide good results for general problems including the text categorization. However, we think that it has an important limitation when applied to the multi-category detection problem because it does not consider an intrinstic nature, *i.e.,* a generative model of multi-category text.

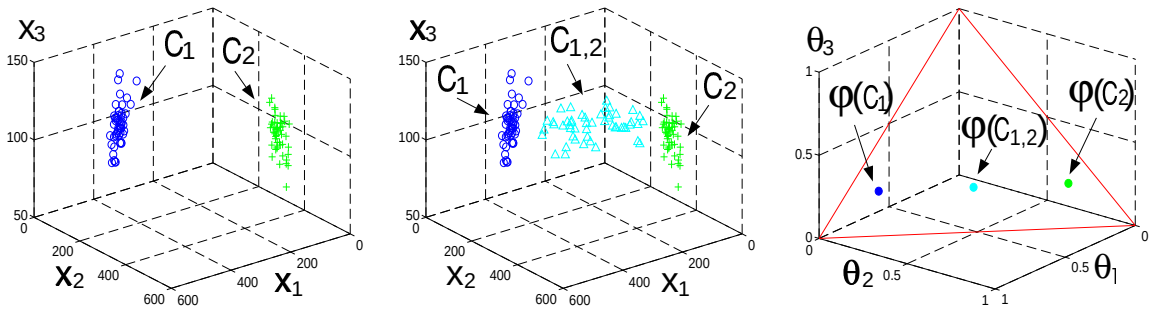
In this paper, we present a novel single-shot approach for the multi-category detection problem using probabilistic generative models, newly proposed in this paper. One might think that the conventional mixture models including a mixture of naive Bayes model [12] would be suitable for this purpose. Unfortunately, however, they are inappropriate for multi-category text modeling because in distributional mixture models, a sample is assumed to be probabilistically generated from *one* of the component distributions. In other words, the conventional distributional mixture models are useful for *hierarchical* or *tree-structured* category representation, but are not appropriate for *multi-* or *network-structured* category representaion.

In contrast, in this paper, we first propose a new type of mixture model, called a *parametric mixture model: PMM-I and PMM-II*. (PMM-II is a more sophisticated version of PMM-I.) Like the naive Bayes model [8][10], PMMs also employ independent word-based representation, known as "bag-of-words (BOW)" representation, which ignores the order of the word occurrence in a document. More specifically, based on the BOW, the word-frequency distribution over the vocaburary is formulated not as a mixture of multinomial distributions, but as a *single* multinomial model. However, a parameter vector of a single multimomial distribution is formulated by a mixture of basis multinomial parameter vectors, and each of the basis vectors corresponds to the parameter vector of a multinomial distribtution for a single-category text.

2. PARAMETRIC MIXTURE MODELS

A document, d^n, can be represented as a word-frequency vector, $\boldsymbol{x}^n = (x_1^n, \ldots, x_V^n)$, where x_i^n denotes frequency of

SIGKDD 02 Edmonton, Alberta, Canada



(a) Single-category (b) Single & multi-category (c) Parameter vectors

Figure 1: Relationship between word-frequency distribution and multinomial parameter vectors.

word w_i occurrence in d^n among the vocabulary $\mathcal{V} =< w_1, \ldots, w_V >$. Here, V is the total number of words in the vocabulary. Thus, in the BOW, each $\boldsymbol{x}$ is a point in V-dimensional Euclidean space and $\boldsymbol{x}$ is assumed to be generated by a multinomial distribution over the words: $p(\boldsymbol{x};\boldsymbol{\theta}) \propto \prod_{i=1}^{V} \theta_i^{x_i}$, where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_V)$ is a model parameter vector and the ith element θ_i denotes a probability that w_i appears and therefore $\theta_i \geq 0$ and $\sum_{i=1}^{V} \theta_i = 1$. We define a category vector $\boldsymbol{y}^n = (y_1^n, \ldots, y_L^n)$ for d^n, where y_l^n takes a value of 1(0) when d^n belongs (does not belong) to the lth category. Here, L is the total number of categories and L categories are assumed to be known. Note that we assume that *at least* one component in $\boldsymbol{y}^n$ takes a value of 1.

Now, let us consider how word-frequency vectors of multi-category text are distributed. As a simple example, suppose that we have two multinomial distributions $\mathcal{M}_1$ and $\mathcal{M}_2$ for category C_1 and C_2, each of which are specified by $\boldsymbol{\varphi}(C_1) = (0.7, 0.1, 0.2)$ and $\boldsymbol{\varphi}(C_2) = (0.1, 0.7, 0.2)$, respectively. This corresponds to the case where $(L, V) = (2, 3)$. Figure 1(a) shows samples $\{\boldsymbol{x}^n\}$ denoted by 'o'('+') which are artificially generated by $\boldsymbol{\varphi}(C_1)$ ($\boldsymbol{\varphi}(C_2)$). The sum of elements of each frequency vector are distributed from about 100 to 800. Clearly, a parameter vector $\boldsymbol{\varphi}$ lies on a two-dimensional simplex (*ie.*, $\theta_1 + \theta_2 + \theta_3 = 1$), shown as a large triangle in Figure 1(c).

Let $C_{1,2}$ denote a multi-category class which belongs to both C_1 and C_2. Then, it may be assumed that text belonging to $C_{1,2}$ has words related to C_1 and C_2. For example, it seems reasonable to assume that a document having two topics such as "sports" and "music" would consist of a mixture of characteristic words related to both topics, where the mixing weight values between these two categories would not always be equal (*i.e.*, 0.5).

According to the above idea, we generated $C_{1,2}$ samples, denoted by '△' in Figure 1(b), by computing the weighted sum of two arbitrarily selected word-frequency vectors each of which belongs to C_1 and C_2, respectively. One can see that samples in $C_{1,2}$ are distributed between C_1 and C_2 samples. The important point to note is that these samples in $C_{1,2}$ can *never* be generated by a mixture of two multinomial distributions $\mathcal{M}_1$ and $\mathcal{M}_2$.

Clearly, the maximum likelihood estimate of $\boldsymbol{\varphi}(C_k)$ is proportional to $\sum_{\boldsymbol{x} \in C_k} \boldsymbol{x}$, for $k = 1$ and 2. Thus, from the above generative process of $C_{1,2}$ samples, one can easily see that a multinomial parameter vector of the $C_{1,2}$ class, denoted by $\boldsymbol{\varphi}(C_{1,2})$, can be approximated by a linear combination of $\boldsymbol{\varphi}(C_1)$ and $\boldsymbol{\varphi}(C_2)$:

$$\boldsymbol{\varphi}(C_{1,2}) \sim \alpha \boldsymbol{\varphi}(C_1) + (1 - \alpha)\boldsymbol{\varphi}(C_2), \tag{1}$$

where $0 < \alpha < 1$ is a mixing weight. Actually, $\boldsymbol{\varphi}(C_{1,2})$ in Figure 1(c) is the maximum likelihood estimate computed by using $C_{1,2}$ samples.

Generalizing the above idea, we can say that the distribution of word-frequency vectors of multi-category text is a multinomial distribution with a parameter vector generated by a mixture of basis parameter vectors, and each basis vector corresponds to a multinomial distribution of a single category. Since in our approach, unlike in the conventional distributional mixture models, the mixing is performed in the parameter space, we call this a "parametric mixture model" in this paper.

2.1 PMM-I

As mentioned above, multi-category text may sometimes be more weighted to one category than to the rest of the categories among multicategories. However, being averaged over all biases, they could be canceled and therefore $\alpha = 0.5$ in Eq. (1) may be reasonable. This motivates us to construct a model called "Parametric Mixture Model Type I: PMM-I".

More generally, a given multi-category text is specified by a single category vector $\boldsymbol{y}$, and its multinomial parameter can be represented by

$$\boldsymbol{\varphi}(\boldsymbol{y}) = \frac{\sum_{l=1}^{L} y_l \boldsymbol{\theta}_l}{\sum_{l'=1}^{L} y_{l'}}, \tag{2}$$

where $\boldsymbol{\theta}_l = (\theta_{l1}, \ldots, \theta_{lV})$ is a multinomial parameter vector[1] of the category C_l.

Thus, the generative model of PMM-I is represented by

$$P(\boldsymbol{x}|\boldsymbol{y};\Theta) \propto \prod_{i=1}^{V} \varphi_i(\boldsymbol{y})^{x_i} = \prod_{i=1}^{V} \left\{ \frac{\sum_{l=1}^{L} y_l \theta_{li}}{\sum_{l'=1}^{L} y_{l'}} \right\}^{x_i}, \tag{3}$$

[1] It should be written $\boldsymbol{\varphi}(C_l)$, but to make notation simple we use $\boldsymbol{\theta}_l$ instead of $\boldsymbol{\varphi}(C_l)$ hereafter.

where $\varphi_i(\boldsymbol{y})$ denotes the ith element in $\boldsymbol{\varphi}(\boldsymbol{y})$. Note that $\sum_{i=1}^{V}\varphi_i(\boldsymbol{y})=1$ holds. In PMM-I, a set of unknown model parameters is $\Theta=\{\boldsymbol{\theta}_l;\ l=1,\ldots,L\}$.

2.2 PMM-II

We also present "Parametric Mixture Model Type II: PMM-II" as a variant, in which α in Eq. (1) also becomes a free parameter. This model is for use in case where the bias weights mentioned above cannot be ignored. In PMM-II a multinomial parameter vector given $\boldsymbol{y}$ is defined by

$$\boldsymbol{\varphi}(\boldsymbol{y})=\frac{\sum_{l=1}^{L}\sum_{m=1}^{L}y_l y_m\boldsymbol{\theta}_{l,m}}{\sum_{l=1}^{L}y_l\sum_{m=1}^{L}y_m}, \tag{4}$$

where $\boldsymbol{\theta}_{l,m}=\alpha_{l,m}\boldsymbol{\theta}_l+\alpha_{m,l}\boldsymbol{\theta}_m$. $\alpha_{l,m}(>0)$ is a mixing parameter that satisfies $\alpha_{l,m}+\alpha_{m,l}=1,\ \forall l,m$. Note that $\boldsymbol{\theta}_{l,l}\equiv\boldsymbol{\theta}_l$ because $\alpha_{l,l}=0.5$. Moreover, $\boldsymbol{\theta}_{l,m}=\boldsymbol{\theta}_{m,l}$ holds. Eq. (4) is well defined so that when sum of all elements in $\boldsymbol{y}$ is 1 (*i.e.,* multi-category case), $\boldsymbol{\varphi}(\boldsymbol{y})$ cannot coincide with a single-category basis vector $\boldsymbol{\varphi}(C_l)$ even if the mixing parameter value is set to 0 or 1. According to Eq. (4), the generative model of PMM-II is given by

$$P(\boldsymbol{x}|\boldsymbol{y};\Theta)\propto\prod_{i=1}^{V}\left\{\frac{\sum_{l=1}^{L}\sum_{m=1}^{L}y_l y_m\theta_{l,m,i}}{\sum_{l=1}^{L}y_l\sum_{m=1}^{L}y_m}\right\}^{x_i}. \tag{5}$$

where $\theta_{l,m,i}=\alpha_{l,m}\theta_{li}+\alpha_{m,l}\theta_{mi}$. In PMM-II, a set of unknown model parameters is $\Theta=\{\boldsymbol{\theta}_l,\ \alpha_{l,m};\ l=1,\ldots,L,\ m=1,\ldots,L\}$.

3. LEARNING AND PREDICTION

3.1 Learning algorithms

Let $\mathcal{D}=\{\boldsymbol{x}^n,\boldsymbol{y}^n;\ n=1,\ldots,N\}$ denote the given traning data. Then, the estimate of model parameter vector Θ is derived by finding the maximum a posteriori estimate:

$$\begin{aligned}\hat{\Theta}_{map} &= \arg\max_{\Theta}P(\Theta|\mathcal{D})\\ &= \arg\max_{\Theta}\{\sum_{n=1}^{N}\log P(\boldsymbol{x}^n|\boldsymbol{y}^n,\Theta)+\log p(\Theta)\}.\end{aligned} \tag{6}$$

This maximization can be solved within the framework of the penalized EM algorithm [3] as shown later.

Each prior distribution over parameter $\boldsymbol{\theta}_l$ and $\alpha_{l,m}$ is represented by *Dirichlet* distributions.

$$P(\theta_{li})\propto\prod_{l=1}^{L}\prod_{i=1}^{V}\theta_{li}^{\xi-1}\quad\text{and}\quad P(\alpha_{l,m})\propto\prod_{l=1}^{L}\prod_{m=1}^{L}\alpha_{l,m}^{\zeta-1}. \tag{7}$$

ξ and ζ are hyperparameters and in this paper we set $\xi=2$ and $\zeta=2$, which is equivalent to *Laplace smoothing* for θ_{li} and $\alpha_{l,m}$, respectively.

3.1.1 Learning algorithm for PMM-I

According to Eq. (6), the objective function for PMM-I is

$$J(\Theta;\mathcal{D})=\mathcal{L}(\Theta;\mathcal{D})+(\xi-1)\sum_{l=1}^{L}\sum_{i=1}^{V}\log\theta_{li}, \tag{8}$$

where $\mathcal{L}(\Theta;\mathcal{D})$ is the log-likelihood term and is given by

$$\mathcal{L}(\Theta;\mathcal{D})=\sum_{n=1}^{N}\sum_{i=1}^{V}x_i^n\log\sum_{l=1}^{L}h_l^n\theta_{li}. \tag{9}$$

Here, we set $h_l^n=y_l^n/\sum_{l'=1}^{L}y_{l'}^n$. Let $\Theta^{(t)}$ be a parameter estimate at step t. Moreover setting

$$g_{li}^n(\Theta)=\frac{h_l^n\theta_{li}}{\sum_{l'=1}^{L}h_{l'}^n\theta_{l'i}}, \tag{10}$$

and noting that $\sum_{l=1}^{L}g_{li}^n(\Theta)=1$, we rewrite Eq. (9) as:

$$\begin{aligned}\mathcal{L}(\Theta;\mathcal{D}) &= \sum_{n=1}^{N}\sum_{i=1}^{V}x_i^n\{\sum_{l=1}^{L}g_{li}^n(\Theta^{(t)})\}\log\Big\{\Big(\frac{h_l^n\theta_{li}}{h_l^n\theta_{li}}\Big)\sum_{l'=1}^{L}h_{l'}^n\theta_{l'i}\Big\}\\ &= \mathcal{F}(\Theta|\Theta^{(t)})-\mathcal{S}(\Theta|\Theta^{(t)}),\end{aligned} \tag{11}$$

where $\mathcal{F}(\Theta|\Theta^{(t)})$ and $\mathcal{S}(\Theta|\Theta^{(t)})$ are defined as follows:

$$\begin{aligned}\mathcal{F}(\Theta|\Theta^{(t)}) &= \sum_{n=1}^{N}\sum_{i=1}^{V}x_i^n\sum_{l=1}^{L}g_{li}^n(\Theta^{(t)})\log h_l^n\theta_{li},\\ \mathcal{S}(\Theta|\Theta^{(t)}) &= \sum_{n=1}^{N}\sum_{i=1}^{V}x_i^n\sum_{l=1}^{L}g_{li}^n(\Theta^{(t)})\log g_{li}^n(\Theta).\end{aligned}$$

Noting that $\mathcal{S}(\Theta|\Theta^{(t)})\le\mathcal{S}(\Theta^{(t)}|\Theta^{(t)})$ holds from Jensen's inequality, if $\mathcal{F}(\Theta|\Theta^{(t)})\ge\mathcal{F}(\Theta^{(t)}|\Theta^{(t)})$ then $\mathcal{L}(\Theta;\mathcal{D})\ge\mathcal{L}(\Theta^{(t)};\mathcal{D})$ from Eq. (11).

Therefore, by maximizing $\mathcal{F}(\Theta|\Theta^{(t)})+(\xi-1)\sum_{l=1}^{L}\sum_{i=1}^{V}\log\theta_{li}$ with respect to Θ, we can increase $J(\Theta;\mathcal{D})$ monotonically on any iteration of parameter update from $\Theta^{(t)}$ to Θ:

$$\text{For } l=1,\ldots,L,\ i=1,\ldots,V,$$
$$\theta_{li}^{(t+1)}=\frac{\sum_{n=1}^{N}x_i^n g_{li}^n(\Theta^{(t)})+\xi-1}{\sum_{i=1}^{V}\sum_{n=1}^{N}x_i^n g_{li}^n(\Theta^{(t)})+V(\xi-1)}. \tag{12}$$

Here g_{li}^n is given by Eq. (10). It is worth mentioning that performing this parameter update, the algorithm always converges to the global optimum of $J(\Theta;\mathcal{D})$ since the Hessian matrix of the objective function is negative definite.

3.1.2 Learning algorithm for PMM-II

The objective function for PMM-II becomes

$$J(\Theta;\mathcal{D})=\mathcal{L}(\Theta;\mathcal{D})+(\xi-1)\sum_{l,i}\log\theta_{l,i}+(\zeta-1)\sum_{l,m}\log\alpha_{l,m}.$$

Here, the log-likelihood term is given by

$$\mathcal{L}(\Theta;\mathcal{D})=\sum_{n=1}^{N}\sum_{i=1}^{V}x_i^n\log\sum_{l=1}^{L}\sum_{m=1}^{L}h_l^n h_m^n\theta_{l,m,i}. \tag{13}$$

Using a similar method to that for PMM-I, we can obtain the following parameter update formulae for θ_{li} and $\alpha_{l,m}$. The results are given below:

$$\text{For } l=1,\ldots,L,\ i=1,\ldots,V,$$
$$\theta_{li}=\frac{2\sum_{n=1}^{N}x_i^n\sum_{m=1}^{L}q_{l,m,i}^n(\Theta^{(t)})\lambda_{l,m,i}(\Theta^{(t)})+\xi-1}{2\sum_{i=1}^{V}\sum_{n=1}^{N}x_i^n\sum_{m=1}^{L}q_{l,m,i}^n(\Theta^{(t)})\lambda_{l,m,i}(\Theta^{(t)})+V(\xi-1)} \tag{14}$$

Table 1: Summary of Detection Problems

Problem Name	V	L	PMC	ANW
Arts & Humanities	23146	26	44.4%	111.1
Business & Economy	21924	30	42.4%	102.1
Computers & Internet	34096	33	30.2%	128.2
Education	27534	33	33.1%	111.8
Entertainment	32001	21	27.7%	145.7
Health	30605	32	46.8%	108.8
Recreation & Sports	30324	22	30.8%	129.9
Reference	39679	33	14.5%	163.7
Science	37187	40	32.0%	173.3
Social & Science	52350	39	21.6%	154.4
Society & Culture	31802	27	40.4%	176.2

For $l = 1, \ldots, L,\ m = 1, \ldots, L$,

$$\alpha_{l,m} = \frac{\sum_{n=1}^{N}\sum_{i=1}^{V} x_i^n q_{l,m,i}^n(\Theta^{(t)})\lambda_{l,m,i}(\Theta^{(t)}) + \zeta - 1}{\sum_{i=1}^{V}\sum_{n=1}^{N} x_i^n q_{l,m,i}^n(\Theta^{(t)}) + V(\zeta - 1)}. \quad (15)$$

Here, $q_{l,m,i}^n(\Theta)$ and $\lambda_{l,m,i}(\Theta)$ are defined by

$$q_{l,m,i}^n(\Theta) = \frac{h_l^n h_m^n \theta_{l,m,i}}{\sum_{l=1}^{L}\sum_{m=1}^{L} h_l^n h_m^n \theta_{l,m,i}}, \quad \lambda_{l,m,i}(\Theta) = \frac{\alpha_{l,m}\theta_{li}}{\theta_{l,m,i}}. \quad (16)$$

Note that $\lambda_{l,m,i} + \lambda_{m,l,i} = 1$ and $q_{l,m,i}^n = q_{m,l,i}^n$ hold.

3.2 Prediction algorithm

Let $\hat{\Theta}$ denote the estimated parameter. Then, applying Bayes' rule, the optimum category vector $\boldsymbol{y}^*$ for $\boldsymbol{x}^*$ of a new sample is defined as: $\boldsymbol{y}^* = \arg\max_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x}^*; \hat{\Theta})$ under a uniform class prior assumption. This maximization problem belongs to the *zero-one integer problem* (*i.e.*,, NP-hard problem). Clearly, an exhaustive search is prohibitive for a large L. To solve this problem, we utilize a simple heuristic greedy-search algorithm. That is, first, only one y_{l_1} value is set to 1 so that $P(\boldsymbol{y}|\boldsymbol{x}^*; \hat{\Theta})$ is maximized. Then, for the rest elements, only one y_{l_2} value is set to 1 with y_{l_1} is fixed. This procedure is repeated until $P(\boldsymbol{y}|\boldsymbol{x}^*; \hat{\Theta})$ cannot increase. Namely, this algorithm successively determines an element in $\boldsymbol{y}$ so as to improve the posterior probability until its value does not improve. This algorithm is of great efficiency because it requires the calculation of the posterior probability at most $L(L+1)/2$ times, while the exhaustive search needs $2^L - 1$ times.

4. APPLICATION

4.1 Multi-topic detection of WWW pages

We designed a series of problems for detecting multiple topics of World Wide Web pages. We focused on the "yahoo.com" domain to collect Web pages because this domain is a famous portal site and most related pages linked from this domain are registered by site recommendation and therefore links may be reliable. Yahoo consists of 14 top-level categories and each of these categories is classified into a number of second-level subcategories. We formalized a multi-category detection problem by regarding each list of the second-level subcategories as categories to be detected. Thus, we can obtain 14 *independent* multi-category detection problems[2].

To collect a set of related Web pages for each problem, we ran a software robot called "*GNU Wget* (version 1.5.3)[3]" 14 times by designating each of the pages linked from the 14 top-level categories as an origin page, where hyperlinks of each page were recursively followed until depth five from each origin page. However, unfortunately for "News & Media", "Government", and "Regional", we could not collect enough number of pages due to problems caused by our communication network security. Thus, we excluded these three problems and the remaining eleven problems shown in Table 1 were used in our experiments[4]. Note that these 11 problems were solved *independently*.

Table 1 shows the statistics of our detection problems. V is the total number of different words in the vocabulary, L is the number of categories, PMC is the percentage of the number of samples that belong to more than a single category, and ANW denotes the averaged number of words used in a page. Clearly, each of the collected WWW pages is represented as a very high dimensional word-frequency vector, but uses at most a few hundreds of words on average.

We compared our PMMs with the convetional methods: naive Bayes (NB), SVM, k-nearest neighbor (kNN), and three-layer neural networks (NN). We used linear SVM*light* (version 4.0) [7] with tuning the C (penalty cost) and J (cost-factor for negative and positive samples) parameters for each binary classification to improve the SVM results. J was set to $|s_l^0|/|s_l^1|$ for every category as suggested in [11]. When performing the SVM, each word-frequency vector $\boldsymbol{x}^n$ was normalized to be $\sum_i x_i^n = 1$.

We employed the cosine similarity for kNN method (see [14] for more details). We tried $k = 1, \ldots, 15$ and selected the best one for test samples to evaluate its potential performance, although it is clearly unfair advantage. As for NN, an NN consists of V input units and L output units for estimating a category vector from each frequency vector. We used 50 hidden units. An NN was trained so as to maximize a sum of cross-entropy functions [1] for target and estimated category vectors of training samples, together with a regularization term consisting of a sum of squared NN weights.

4.2 Performance Measures

In the case of the multi-category detection problem, the standard *accuracy* measure is inappropriate because a high accuracy can be achieved by always predicting negative (0) values. We used the *F-measure* as the performance measure instead [9]. The F-measure is defined as the weighted harmonic average of two well-known statistics, *precision*, P, and *recall*, R, widely used in information retrieval.

[2]Of course, it may be natural to consider a 14 top-category detection problem. However, in this case since the number of categories seems small (just 14) and only one set detection problem can be obtained, we used second-level categories.

[3]One can download from "ftp://ftp.gnu.org/pub/gnu/wget".

[4]Note that since our data collection was performed during early 2001, the second-level subcategories used in our experiments are slightly different from the current ones.

Table 2: Detection performance for test samples using $2,000$ training samples.

Problem Name	NB	SVM	kNN	NN	PMM1	PMM2
Arts&Humanities	41.6 (1.9)	47.1 (0.3)	40.0 (1.1)	43.3 (0.2)	**50.6** (1.0)	48.6 (1.0)
Business&Economy	75.0 (0.6)	74.5 (0.8)	**78.4** (0.4)	77.4 (0.5)	75.5 (0.9)	72.1 (1.2)
Computers&Internet	56.5 (1.3)	56.2 (1.1)	51.1 (0.8)	53.8 (1.3)	**61.0** (0.4)	59.9 (0.6)
Education	39.3 (1.0)	47.8 (0.8)	42.9 (0.9)	44.1 (1.0)	**51.3** (2.8)	48.3 (0.5)
Entertainment	54.5 (0.8)	56.9 (0.5)	47.6 (1.0)	54.9 (0.5)	**59.7** (0.4)	58.4 (0.6)
Health	66.4 (0.8)	**67.1** (0.3)	60.4 (0.5)	66.0 (0.4)	66.2 (0.5)	65.1 (0.3)
Recreation	51.8 (0.8)	52.1 (0.8)	44.4 (1.1)	49.6 (1.3)	**55.2** (0.5)	52.4 (0.6)
Reference	52.6 (1.1)	55.4 (0.6)	53.3 (0.5)	55.0 (1.1)	**61.1** (1.4)	60.1 (1.2)
Science	42.4 (0.9)	49.2 (0.7)	43.9 (0.6)	45.8 (1.3)	**51.4** (0.7)	49.9 (0.8)
Social&Science	41.7 (10.7)	**65.0** (1.1)	59.5 (0.9)	62.2 (2.3)	62.0 (5.1)	56.4 (6.3)
Society&Culture	47.2 (0.9)	51.4 (0.6)	46.4 (1.2)	50.5 (0.4)	**54.2** (0.2)	52.5 (0.7)

Let $\boldsymbol{y}^n = (y_1^n, \ldots, y_L^n)$ and $\hat{\boldsymbol{y}}^n = (\hat{y}_1^n, \ldots, \hat{y}_L^n)$ be an actual and predicted category vectors for $\boldsymbol{x}^n$. Then, in our case, P and R per each sample can be computed as

$$P_n = \frac{\sum_{l=1}^L y_l^n \hat{y}_l^n}{\sum_{l=1}^L \hat{y}_l^n}, \quad R_n = \frac{\sum_{l=1}^L y_l^n \hat{y}_l^n}{\sum_{l=1}^L y_l^n} \tag{17}$$

Using Eq. (17), we can obtain $F_n = 2P_nR_n/(P_n + R_n)$. Finally, we compute $\bar{F}$ averaged over all N test samples: $\bar{F} = \frac{1}{N}\sum_{n=1}^N F_n$.

4.3 Results and Discussion

We did not perform any feature transformation such as TFIDF (see, *e.g.,* [14]) because we wanted to purely evaluate the basic performance of each detection method. For every detection problem, we evaluated all the methods mentioned above by using five pairs of training and test sample sets.

In our first set of experiments, the number of training (test) samples was set to $2,000$ ($3,000$). In our second set of experiments, we focused on detection performance for test samples with *unseen* category vectors that did not appear in the training samples. For convenience, hereafter we call this test data *uc-test* samples to discriminate between the two kinds of test data. Since uc-test samples always had multiple topics and their frequency vectors were substantially different from those in the training samples, they are available to severely evaluate the generalization ability of each of the detection methods. In our third set of experiments, to evaluate the robustness of the PMM approach, we reduced the number of training samples from $2,000$ to 500.

We compare the mean of $\bar{F}$ values over five trials on the test (Tables 2 and 4) or the uc-test (Tables 3 and 5) samples. Moreover, the standard deviation of the five trials is also shown in each parenthesis. For the lack of space, we omitted the standard deviations in Tables 3 and 5. PMMs took about five minutes for trainig (2,000 data) and about just one minute for test (3,000 data) on 2.0 Ghz pentium, averaged over 11 problmes.

By tuning parameters, SVM produced fairly better results than the NB method, although SVM is also binary approach. The detection performance by SVM, however, were lower than those by PMMs in almost all problems. These experimental results support our claim that since SVM does not consider generative models of multi-category text, it has an important limitation when applied to the multi-category detection problem.

Table 3: Detection performance for uc-test samples using $2,000$ training samples.

Problem	NB	SVM	kNN	NN	PMM1	PMM2
Arts.	22.4	26.9	26.0	22.1	**31.1**	31.0
Busi.	32.5	36.4	39.3	37.9	**39.5**	38.7
Comp.	24.7	27.8	30.3	25.8	**33.3**	33.6
Edu.	17.9	24.6	24.6	23.3	**31.0**	30.6
Enter.	24.6	29.8	28.9	25.0	**37.7**	37.7
Health	37.0	**40.8**	40.1	37.8	40.1	40.2
Rec.	27.6	29.2	29.1	26.4	**35.7**	35.2
Ref.	20.5	24.9	26.9	23.6	32.4	**33.2**
Sci.	22.7	28.5	28.3	24.6	**34.5**	33.8
Soc.&Sci.	20.8	25.9	28.4	22.4	**29.8**	29.3
Soc.& Cul.	22.5	26.5	28.0	26.0	33.2	**33.5**

When the trainig sample size was 2,000, kNN provided comparable results to the NB method. On the other hand, when the trainig sample size was 500, it obtained comparable or slightly better results than SVM. However, in both cases, PMMs significantly outperformed kNN. This indicates that perhaps memory-based approach has a limitation of its generalization ability and it is of quite importance to extract the smaller number of representative prototypes like in our approach.

The results of *well-regularized* NN were moderate, although it can make curved discrimination boundaries. This means that such kind of flexibility would be unnecessary for discrimination of high-dimensional, sparse text samples even in the case of multi-category detection problem. In additon, the training time of NN was intolerable.

In our experiments, PMM-I provided better results than PMM-II, although there were no significant differences in the case of uc-test samples. This indicates that a model with fixed $\alpha_{l,m} = 0.5$ seems sufficient at least for the WWW pages used in the experiments. In the case of "Social & Science", since the standard deviation was also relatively large (5.1), we examined the result in more detail and found that unfortunately since $\xi = 2$ was inappropriate for three trials among five ones, the performance of PMM-I could be improved by tuning the hyperparameter. Incidentally, the best performance for this problem was obtained by SVM.

Table 4: Detection performance for test samples using 500 training samples.

Problem Name	NB	SVM	kNN	NN	PMM1	PMM2
Arts&Humanities	21.2 (1.0)	32.5 (0.5)	34.7 (0.4)	33.8 (0.4)	**43.9** (1.0)	43.2 (0.8)
Business&Economy	73.9 (0.7)	73.8 (1.2)	**75.6** (0.6)	74.8 (0.9)	75.2 (0.4)	69.7 (8.9)
Computers&Internet	46.1 (2.9)	44.9 (1.9)	44.1 (1.2)	45.1 (1.0)	**56.4** (0.3)	55.4 (0.5)
Education	15.2 (0.9)	33.6 (0.5)	37.1 (1.0)	33.8 (1.1)	41.8 (1.2)	**41.9** (0.7)
Entertainment	34.1 (1.6)	42.7 (1.3)	43.9 (1.0)	45.3 (0.9)	53.0 (0.3)	**53.1** (0.6)
Health	50.2 (0.3)	56.0 (1.0)	54.4 (0.9)	57.2 (0.7)	58.9 (0.9)	**59.4** (1.0)
Recreation	22.1 (0.8)	32.1 (0.5)	37.4 (1.1)	33.9 (0.8)	**46.5** (1.3)	45.5 (0.9)
Reference	32.7 (4.4)	38.8 (0.6)	48.1 (1.3)	43.1 (1.0)	**54.1** (1.5)	53.5 (1.5)
Science	17.6 (1.6)	32.5 (1.0)	35.3 (0.4)	31.6 (1.7)	40.3 (0.7)	**41.0** (0.5)
Social&Science	40.6 (12.3)	55.0 (1.1)	53.7 (0.6)	55.8 (4.0)	57.8 (6.5)	**57.9** (5.9)
Society&Culture	34.2 (2.2)	38.3 (4.7)	40.2 (0.7)	40.9 (1.2)	**49.7** (0.9)	49.0 (0.5)

Table 5: Detection performance for uc-test samples using 500 training samples.

Problem	NB	SVM	kNN	NN	PMM1	PMM2
Arts.	9.7	18.4	24.9	18.7	27.8	**28.9**
Busi.	30.3	33.7	**35.2**	33.4	32.6	34.0
Comp.	17.8	15.9	24.9	18.6	30.3	**31.1**
Edu.	5.7	16.4	20.0	15.5	25.5	**26.8**
Enter.	11.9	19.8	25.2	21.6	29.6	**30.6**
Health	22.8	29.8	**34.9**	29.4	32.7	33.9
Rec.	10.4	15.0	23.7	14.5	28.6	**30.1**
Ref.	7.3	13.6	22.9	15.5	27.0	**27.7**
Sci.	5.2	14.6	21.8	11.9	25.1	**25.9**
Soc.&Sci.	12.9	16.8	22.8	17.7	23.9	**26.6**
Soc.&Cul.	12.4	17.2	24.4	18.3	27.2	**28.3**

5. CONCLUSING REMARKS

In this paper we have presented a novel approach based on parametric mixture models for multi-topic detection of text, and efficient algorithms for both learning and prediction. Clearly, some work remains in order to extend our approach and algorithms, and to evaluate them by using a wider variety of problems. Neverthless, we have taken some important steps along the path, and we are encouraged by our current results on the important problem of detecting multiple topics of real World Wide Web pages.

Recently, sophisticated distributional mixture models for text modeling, *pLSI/aspect model* [5] and *Latent Dirichlet Allocation Model* [2], have been proposed to represent several *latent* subtopics. Combine these models with our models might be interesting.

6. REFERENCES

[1] C. Bishop. *Neural networks for pattern recognition*. Clarendon Press, Oxfor, 1996.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. to appear in *Advances in Neural Information Processing Systems 14 (NIPS14)*. MIT Press.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1-38. 1977.

[4] S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proc. of ACM-CIKM'98*. 1998.

[5] T. Hofmann. Probabilistic latent semantic indexing. In *Proc. of the Twenty-Second Annual International SIGIR Conference (SIGIR'99)*. 1999.

[6] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proc. of the European Conference on Machine Learning (ECML'98)*. 137-142, Berlin, 1998.

[7] T. Joachims. SVM light (version 4). http://ais.gmd.de/thorsten.

[8] D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. In *Third Anual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, 81-93. 1994.

[9] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*, MIT press, Cambridge, 1999.

[10] A. McCallum and K. Nigam. A comparison of event models for naive Bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, Tech. Rep. WS-98-05. 1998.

[11] K. Morik, P. Brockhausen, and T. Joachims. Combining statistical learning with knowledge-based approach. A case study in intensive care monitoring. In *Proc. of International Conference on Machine Learning (ICML'99)*, 1999.

[12] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39, 103-134. 2000.

[13] G. Salton. *Automatic Text Processing*. Addison-Wesley, 1988.

[14] Y. Yang and J. Pederson. A comparative study on feature selection in text categorization. In *Proc of International Conference on Machine Learning (ICML'97)*, 412-420. 1997.

[15] V. N. Vapnik. *Statistical learning theory*. John Wiley & Sons, Inc., New York. 1998.