Available online at www.sciencedirect.com
ScienceDirect
Information Processing and Management 43 (2007) 379–392
INFORMATION PROCESSING & MANAGEMENT
www.elsevier.com/locate/infoproman

# A hybrid generative/discriminative approach to text classification with additional information

Akinori Fujino *, Naonori Ueda, Kazumi Saito

*NTT Communication Science Laboratories, NTT Corporation, 2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan*

Received 27 May 2006; accepted 25 July 2006
Available online 11 October 2006

**Abstract**

This paper presents a classifier for text data samples consisting of main text and additional components, such as Web pages and technical papers. We focus on multiclass and single-labeled text classification problems and design the classifier based on a hybrid composed of probabilistic generative and discriminative approaches. Our formulation considers individual component generative models and constructs the classifier by combining these trained models based on the maximum entropy principle. We use naive Bayes models as the component generative models for the main text and additional components such as titles, links, and authors, so that we can apply our formulation to document and Web page classification problems. Our experimental results for four test collections confirmed that our hybrid approach effectively combined main text and additional components and thus improved classification performance.


*Keywords:* Multiclass and single-labeled text classification; Multiple components; Maximum entropy principle; Naive Bayes model

## 1. Introduction

Text data samples such as Web pages and technical papers usually contain multiple components. For example, Web pages consist of main text and additional components such as titles, hyperlinks, anchor text, and images. Although the main text plays an important role when designing a classifier, additional components may contain substantial information for classification. Therefore, designing classifiers for dealing with multiple components is an important and challenging research issue in the field of machine learning. Recently, such classifiers have been developed for multiple components such as text and hyperlinks on Web pages (Chakrabarti, Dom, & Indyk, 1998; Cohn & Hofmann, 2001; Lu & Getoor, 2003; Sun, Lim, & Ng, 2002), text and citations in papers (Cohn & Hofmann, 2001; Lu & Getoor, 2003), and text and music (Brochu & Freitas, 2003). In this paper, we focus on probabilistic approaches to designing text classifiers that can deal with arbitrary additional components as studied in (Brochu & Freitas, 2003; Lu & Getoor, 2003).

* Corresponding author.
*E-mail addresses:* a.fujino@cslab.kecl.ntt.co.jp (A. Fujino), ueda@cslab.kecl.ntt.co.jp (N. Ueda), saito@cslab.kecl.ntt.co.jp (K. Saito).

0306-4573/$ - see front matter © 2006 Published by Elsevier Ltd.
doi:10.1016/j.ipm.2006.07.013

Existing probabilistic approaches are generative, discriminative, and a hybrid of the two. Generative classifiers learn the joint probability model, $p(\boldsymbol{x}, y)$, of input $\boldsymbol{x}$ and class label $y$, compute $P(y|\boldsymbol{x})$ by using the Bayes rule, and then take the most probable label $y$. However, such direct modeling is hard for arbitrary components consisting of completely different types of media. In Brochu and Freitas (2003), under the assumption of the class conditional independence of all components, the class conditional probability density $p(\boldsymbol{x}^j|y)$ for each component is individually modeled, where $\boldsymbol{x}^j$ stands for the feature vector corresponding to the $j$th component. Hence, as described later, the joint probability density is expressed by the simple product of $p(\boldsymbol{x}^j|y)$.

Discriminative classifiers directly model class posterior probability $P(y|\boldsymbol{x})$ and learn mapping from $\boldsymbol{x}$ to $y$. Multinomial logistic regression (Hastie, Tibshirani, & Friedman, 2001) can be used for this purpose. However, such modeling without consideration of components may have an intrinsic limitation in terms of achieving good classification performance. In Lu and Getoor (2003), a class posterior probability $P(y|\boldsymbol{x}^j)$ for each component is individually modeled, and then the simple product of $P(y|\boldsymbol{x}^j)$ is used for predicting the class to which $\boldsymbol{x}$ belongs.

Hybrid classifiers learn a class conditional probability model for each component, $p(\boldsymbol{x}^j|y)$, and directly model class posterior probability $P(y|\boldsymbol{x})$ by using component generative models. Namely, each component model is estimated on the basis of a generative approach, while the classifier is constructed on the basis of a discriminative approach. Hybrid classifiers are constructed by combining the component generative models with weights determined discriminatively. This contrasts with pure generative and discriminative classifiers, which are based on the simple product of component models without weights. For *binary* classification problems, such a hybrid classifier has already been proposed and applied to documents consisting of two text components (“subject” and “body”) (Raina, Shen, Ng, & McCallum, 2004). It has been shown experimentally that this hybrid classifier achieves higher accuracy than pure generative and discriminative classifiers.

We present a new hybrid classifier for *multiclass* and single-labeled text classification problems. More specifically, we design individual component generative models $p(\boldsymbol{x}^j|y)$ for main text and additional components. Then, by combining the trained component generative models based on the *maximum entropy* (ME) principle (Berger, Della Pietra, & Della Pietra, 1996), we design a class posterior probability distribution $P(y|\boldsymbol{x})$, where a combination weight is provided per component. We expect the way in which the components are combined to utilize additional information effectively and thus improve classification performance.

According to the ME principle, we can obtain another classifier formulation based on a combination of component generative models, where individual combination weights of components are provided per class. Since the different way in which components are combined would affect classification performance, we also explore the formulation.

To enable us to apply our hybrid classifier to documents and Web pages containing main text and additional components such as titles, authors, and hyperlinks, we employ naive Bayes (NB) models as their individual component generative models. We train the NB models of components with a leave-one-out cross-validation of the training samples to improve their generalization abilities.

The organization of the paper is as follows. In Section 2, we review the formulas for conventional generative, discriminative, and hybrid classifiers that deal with main text and additional components. In Section 3, we present the formulation for our hybrid classifier and the method for applying the hybrid classifier to document and Web page classification. In Section 4, our hybrid classifier is evaluated experimentally using four test collections. Our experimental results show the effect of dealing with additional information and of our hybrid approach on the performance of multiclass classification. Related work is reviewed in Section 5, and our conclusions are presented in Section 6.

## 2. Conventional approaches

In multiclass and single-labeled classification problems, a classifier categorizes a feature vector $\boldsymbol{x}$ into one of $K(>2)$ classes $y \in \{1, \ldots, k, \ldots, K\}$. Each feature vector consists of $J$ separate components as $\boldsymbol{x} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^j, \ldots, \boldsymbol{x}^J\}$. The classifier is trained on training sample set $D = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$. In the following, we derive basic formulas for the conventional approaches.

### 2.1. Generative approach

Generative classifiers model joint probability density $p(\boldsymbol{x}, y)$. However, as mentioned above, such direct modeling is hard for arbitrary components that consist of completely different types of media. Under the assumption of the class conditional independence of all components, the joint probability density can be modeled as

$$p(\boldsymbol{x}, y = k; \boldsymbol{\theta}_k) = P(k) \prod_{j=1}^{J} p(\boldsymbol{x}^j | k; \boldsymbol{\theta}_k^j), \tag{1}$$

where $\boldsymbol{\theta}_k^j$ is a model parameter for the $j$th component in the $k$th class and $\boldsymbol{\theta}_k = \{\boldsymbol{\theta}_k^j\}_{j=1}^J$. Note that the component generative model $p(\boldsymbol{x}^j | k; \boldsymbol{\theta}_k^j)$ should be selected according to the features of the component: for example, a multinomial model for text information (Nigam, McCallum, Thrun, & Mitchell, 2000) or a Gaussian model for continuous feature vectors.

Model parameter set $\Theta = \{\boldsymbol{\theta}_k^j\}_{j,k}$ is computed by maximizing the posterior $p(\Theta|D)$ (MAP estimation). According to the Bayes rule, $p(\Theta|D) \propto p(D|\Theta)p(\Theta)$, the objective function for MAP estimation is given by

$$J(\Theta) = \sum_{n=1}^{N} \left\{ \log P(y_n) + \sum_{j=1}^{J} \log p(\boldsymbol{x}_n^j | y_n; \boldsymbol{\theta}_{y_n}^j) \right\} + \sum_{j=1}^{J} \sum_{k=1}^{K} \log p(\boldsymbol{\theta}_k^j). \tag{2}$$

Here, $p(\boldsymbol{\theta}_k^j)$ is a prior over parameter $\boldsymbol{\theta}_k^j$. Clearly, component model parameter $\boldsymbol{\theta}_k^j$ can be optimized without considering the other parameters.

According to the Bayes rule, class posterior probabilities $P(y = k|\boldsymbol{x}; \Theta)$ can be derived as

$$P(y = k|\boldsymbol{x}; \Theta) = \frac{P(k) \prod_{j=1}^{J} p(\boldsymbol{x}^j | k; \boldsymbol{\theta}_k^j)}{\sum_{k'=1}^{K} P(k') \prod_{j=1}^{J} p(\boldsymbol{x}^j | k'; \boldsymbol{\theta}_{k'}^j)}, \quad \forall k. \tag{3}$$

The class label $y$ of $\boldsymbol{x}$ is determined as the $k$ that maximizes $P(k|\boldsymbol{x}; \Theta)$. Note that since the denominator of Eq. (3) is the same for all $k$, it can be approximately predicted from the simple product of $p(\boldsymbol{x}^j | k; \boldsymbol{\theta}_k^j)$.

### 2.2. Discriminative approach

Discriminative classifiers directly model class posterior probabilities $P(y|\boldsymbol{x})$ for all classes. With multinomial logistic regression (MLR) (Hastie et al., 2001), class posterior probabilities are modeled as

$$P(y = k|\boldsymbol{x}; W) = \frac{\exp(\boldsymbol{w}_k \cdot \boldsymbol{x})}{\sum_{k'=1}^{K} \exp(\boldsymbol{w}_{k'} \cdot \boldsymbol{x})}, \quad \forall k, \tag{4}$$

where $W = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K\}$ is a set of unknown model parameters. $\boldsymbol{w}_k \cdot \boldsymbol{x}$ represents the inner product of $\boldsymbol{w}_k$ and $\boldsymbol{x}$. $W$ is estimated for maximizing the following penalized conditional log-likelihood:

$$J(W) = \sum_{n=1}^{N} \log P(y_n | \boldsymbol{x}_n; W) + \log p(W). \tag{5}$$

Here, $p(W)$ is a prior over parameter $W$.

In Lu and Getoor (2003), an individual MLR model $P(k|\boldsymbol{x}^j; W^j)$ is designed for each component, and the estimate $\hat{W}^j$ of the model parameter is independently computed. The class label $y$ of $\boldsymbol{x}$ is determined as the $k$ that maximizes the product of the class posterior probabilities estimated in the components such that

$$y = \arg\max_k \prod_{j=1}^{J} P(k|\boldsymbol{x}^j; \hat{W}^j). \tag{6}$$

### 2.3. Hybrid approach for binary classification

Hybrid classifiers learn a class conditional probability model for each component, $p(\boldsymbol{x}^j|y)$, and directly model the class posterior probability $P(y|\boldsymbol{x})$ by using the trained component models. In Raina et al. (2004), binary classifiers are derived as follows. The class posterior probability in Eq. (3) is equivalently transformed to

$$P(y=1|\boldsymbol{x};\Theta)=\frac{1}{1+\exp\left\{\sum_{j=1}^{J}\log\frac{p(\boldsymbol{x}^j|y=2;\boldsymbol{\theta}_2^j)}{p(\boldsymbol{x}^j|y=1;\boldsymbol{\theta}_1^j)}+\log\frac{P(y=2)}{P(y=1)}\right\}}. \tag{7}$$

Then, by introducing the weight parameters $b_j$ for the components and $b_0=\log\{P(y=2)/P(y=1)\}$, the class posterior probability is extended as follows:

$$R(y=1|\boldsymbol{x};\Theta,B)\equiv\frac{1}{1+\exp\left\{\sum_{j=1}^{J}b_j\log\frac{p(\boldsymbol{x}^j|y=2;\boldsymbol{\theta}_2^j)}{p(\boldsymbol{x}^j|y=1;\boldsymbol{\theta}_1^j)}+b_0\right\}}. \tag{8}$$

Here, the class posterior probability for $y=2$ is provided as $R(y=2|\boldsymbol{x};\Theta,B)=1-R(y=1|\boldsymbol{x};\Theta,B)$. The weight parameter set $B=\{b_j\}_{j=0}^{J}$ is estimated as the parameter of logistic regression, according to the maximum class posterior likelihood as mentioned above.

## 3. Proposed method

In this section, we present the formulation for our hybrid classifier and a method for applying the hybrid classifier to document and Web page classification.

### 3.1. Hybrid approach

#### 3.1.1. Component generative models

In our formulation, we simply design individual component generative models without strictly assuming class conditional independence as described in Section 2.1. Let $p(\boldsymbol{x}^j|k;\boldsymbol{\theta}_k^j)$ be the $j$th component generative model in the $k$th class, where $\boldsymbol{\theta}_k^j$ denotes the model parameter. $\boldsymbol{\theta}_k^j$ is computed using MAP estimation. The $\boldsymbol{\theta}_k^j$ estimate is computed to maximize the objective function using training sample set $D$.

#### 3.1.2. Discriminative class posterior design

We provide class posterior probabilities based on the weighted combination of the component generative models to improve the classification performance. More specifically, we design a class posterior probability distribution by combining component generative models based on the maximum entropy (ME) principle (Berger et al., 1996).

The ME principle is a framework for obtaining a probability distribution, which prefers the most uniform models that satisfy any given constraints. Let $R(k|\boldsymbol{x})$ be a target distribution that we wish to specify using the ME principle. A constraint is that the expectation of log-likelihood with respect to the target distribution $R(k|\boldsymbol{x})$ is equal to the expectation of log-likelihood with respect to the empirical distribution $\tilde{p}(\boldsymbol{x},k)=\sum_{n=1}^{N}\delta(\boldsymbol{x}-\boldsymbol{x}_n,k-y_n)/N$ of the training samples as

$$\sum_{\boldsymbol{x},k}\tilde{p}(\boldsymbol{x},k)\log p(\boldsymbol{x}^j|k;\hat{\boldsymbol{\theta}}_k^j)=\sum_{\boldsymbol{x},k}\tilde{p}(\boldsymbol{x})R(k|\boldsymbol{x})\log p(\boldsymbol{x}^j|k;\hat{\boldsymbol{\theta}}_k^j),\quad\forall j, \tag{9}$$

where $\tilde{p}(\boldsymbol{x})=\sum_{n=1}^{N}\delta(\boldsymbol{x}-\boldsymbol{x}_n)/N$ is the empirical distribution of $\boldsymbol{x}$, and $\hat{\boldsymbol{\theta}}_k^j$ represents the estimate of component generative model parameter $\boldsymbol{\theta}_k^j$. We also restrict $R(k|\boldsymbol{x})$ so that it has the same class probability as seen in the training data, such that

$$\sum_{\boldsymbol{x}}\tilde{p}(\boldsymbol{x},k)=\sum_{\boldsymbol{x}}\tilde{p}(\boldsymbol{x})R(k|\boldsymbol{x}),\quad\forall k. \tag{10}$$

By maximizing the conditional entropy $H(R) = -\sum_{\boldsymbol{x},k}\tilde{p}(\boldsymbol{x})R(k|\boldsymbol{x})\log R(k|\boldsymbol{x})$ under these constraints, we can obtain the target distribution:

$$R(k|\boldsymbol{x};\hat{\Theta},\Lambda) = \frac{e^{\mu_k}\prod_{j=1}^{J}p(\boldsymbol{x}^j|k;\hat{\boldsymbol{\theta}}_k^j)^{\lambda_j}}{\sum_{k'=1}^{K}e^{\mu_{k'}}\prod_{j=1}^{J}p(\boldsymbol{x}^j|k';\hat{\boldsymbol{\theta}}_{k'}^j)^{\lambda_j}}, \quad \forall k, \tag{11}$$

where $\Lambda = \{\{\lambda_j\}_{j=1}^{J}, \{\mu_k\}_{k=1}^{K}\}$ is a set of Lagrange multipliers. $\lambda_j$ provides a combination weight for the $j$th component generative model, and $\mu_k$ provides a bias for the $k$th class. The distribution $R(k|\boldsymbol{x};\hat{\Theta},\Lambda)$ gives us the formulation of a discriminative classifier that consists of component generative models. In this paper, we call this classifier ''Hybrid''.

We can regard the distribution $R(k|\boldsymbol{x};\hat{\Theta},\Lambda)$ derived from the ME principle as a natural extension of the class posterior $P(k|\boldsymbol{x};\Theta)$ shown in Eq. (3). Actually, if $\lambda_j = 1$, $\forall j$ and $e^{\mu_k} = P(k), \forall k$, $R(k|\boldsymbol{x};\hat{\Theta},\Lambda)$ is reduced to $P(k|\boldsymbol{x};\Theta)$. We can also regard $R(k|\boldsymbol{x};\hat{\Theta},\Lambda)$ shown in Eq. (11) as a natural extension of $R(k|\boldsymbol{x};\Theta,B)$ for the binary classifications shown in Eq. (8). If $K = 2$, $\lambda_j = b_j$, and $\mu_2 - \mu_1 = b_0$, $R(k|\boldsymbol{x};\hat{\Theta},\Lambda)$ is reduced to $R(k|\boldsymbol{x};\Theta,B)$.

According to the ME principle, the solution of $\Lambda$ in Eq. (11) is equal to $\Lambda$ that maximizes the log likelihood for $R(k|\boldsymbol{x};\hat{\Theta},\Lambda)$ of training samples $(\boldsymbol{x}_n, y_n) \in D$ (Berger et al., 1996; Nigam, Lafferty, & McCallum, 1999). However, $D$ is also used to estimate $\Theta$. Using the same training samples for $\Lambda$ as $\Theta$ may lead to a bias estimation of $\Lambda$. Thus, a leave-one-out cross-validation of the training samples is used for estimating $\Lambda$ (Raina et al., 2004). Let $\hat{\Theta}^{(-n)}$ be the generative model parameter estimated by using all the training samples except $(\boldsymbol{x}_n, y_n)$. The objective function of $\Lambda$ then becomes

$$J(\Lambda) = \sum_{n=1}^{N}\log R(y_n|\boldsymbol{x}_n;\hat{\Theta}^{(-n)},\Lambda) + \log p(\Lambda), \tag{12}$$

where $p(\Lambda)$ is a prior over parameter $\Lambda$. We use a Gaussian prior (Chen & Rosenfeld, 1999) as

$$p(\Lambda) \propto \prod_{j=1}^{J}\exp\left(-\frac{(\lambda_j-1)^2}{2\sigma_j^2}\right)\prod_{k=1}^{K}\exp\left(-\frac{\mu_k^2}{2\rho_k^2}\right). \tag{13}$$

We can compute an estimate of $\Lambda$ to maximize $J(\Lambda)$ by using the L-BFGS algorithm (Liu & Nocedal, 1989), which is a quasi-Newton method. In this computation, a global convergence is guaranteed, since $J(\Lambda)$ is an upper convex function. We summarize the algorithm for estimating these model parameters in Fig. 1.

### 3.1.3. Another class posterior by ME

According to the ME principle, we can also obtain the class posterior distribution based on a hybrid of the generative models and multinomial logistic regression as

$$R(k|\boldsymbol{x};\hat{\Theta},\Lambda) = \frac{\exp\left\{\sum_{j=1}^{J}\lambda_{jk}\log p(\boldsymbol{x}^j|k;\hat{\boldsymbol{\theta}}_k^j) + \mu_k\right\}}{\sum_{k'=1}^{K}\exp\left\{\sum_{j=1}^{J}\lambda_{jk'}\log p(\boldsymbol{x}^j|k';\hat{\boldsymbol{\theta}}_{k'}^j) + \mu_{k'}\right\}}, \quad \forall k, \tag{14}$$

Given training sample set: $D = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$
1. Compute $\hat{\Theta}$ using Eq. (2).
2. Compute $\hat{\Theta}^{(-n)}$, $\forall n$ by applying Eq. (2) to training samples except $(\boldsymbol{x}_n, y_n)$.
3. Compute $\Lambda$ using Eq. (12) under fixed $\hat{\Theta}^{(-n)}$.
4. Output a classifier $R(k|\boldsymbol{x};\hat{\Theta},\hat{\Lambda})$.

Fig. 1. Algorithm for learning model parameters.

by using the constraint:

$$\sum_{\boldsymbol{x}} \tilde{p}(\boldsymbol{x},k)\log p(\boldsymbol{x}^j|k;\hat{\boldsymbol{\theta}}_k^j) = \sum_{\boldsymbol{x}} \tilde{p}(\boldsymbol{x})R(k|\boldsymbol{x})\log p(\boldsymbol{x}^j|k;\hat{\boldsymbol{\theta}}_k^j), \quad \forall j,\forall k, \tag{15}$$

instead of Eq. (9). Here, $\Lambda = \{\{\lambda_{jk}\}_{j,k},\{\mu_k\}_k\}$ in Eq. (14) is a set of Lagrange multipliers. $\lambda_{jk}$ provides a combination weight for the $j$th component generative model in the $k$th class. Namely, an individual combination weight of the $j$th component is provided for each class. This contrasts with the class posterior distribution for Hybrid shown in Eq. (11), where the same combination weight $\lambda_j$ is provided for all the classes. Since $\lambda_{jk}$ is discriminatively determined as well as $\lambda_j$, we can regard $R(k|\boldsymbol{x};\hat{\Theta},\Lambda)$ shown in Eq. (14) as another hybrid classifier formulation based on a discriminative combination of component generative models. In this paper, we call this classifier "Hybrid-L".

We assume that Hybrid-L is better fitted to training samples than Hybrid, because there are more combination weight parameters in Hybrid-L. This may result in Hybrid-L exhibiting excellent classification performance. Therefore, we compare the classification performance of Hybrid-L with that of Hybrid experimentally in Section 4.

### 3.2. Application to text classification

We apply the hybrid classifiers, Hybrid and Hybrid-L, to text data samples consisting of main text and additional information such as link (citation) and author information. For text information, we employ naive Bayes (NB) models (Nigam et al., 2000) as component generative models using an independent word-based representation, known as the Bag-of-Words (BOW) representation. Let $\boldsymbol{x}^j = (x_1^j,\ldots,x_i^j,\ldots,x_{V_j}^j)$ represent the feature (word-frequency) vector of the $j$th component of a data sample, where $x_i^j$ denotes the frequency of the $i$th word in the $j$th component and $V_j$ denotes the number of vocabulary words included in the $j$th component. In the NB model, the probability distribution of $\boldsymbol{x}^j$ in the $k$th class is regarded as a multinomial distribution:

$$P(\boldsymbol{x}^j|k;\boldsymbol{\theta}_k^j) \propto \prod_{i=1}^{V_j}(\theta_{ki}^j)^{x_i^j}. \tag{16}$$

Here, $\theta_{ki}^j > 0$ and $\sum_{i=1}^{V_j}\theta_{ki}^j = 1$. $\theta_{ki}^j$ is the probability that the $i$th word appears in the $j$th component of a text data sample belonging to the $k$th class.

We also employ NB models as component generative models for link and author information, using a feature vector $\boldsymbol{x}^j$ denoting the frequencies of features (links or authors). See Section 4.1 for details of the features for link and author information.

Using a feature vector $\tilde{\boldsymbol{x}} = \{\tilde{\boldsymbol{x}}^1,\ldots,\tilde{\boldsymbol{x}}^j,\ldots,\tilde{\boldsymbol{x}}^J\}$ normalized with vector size $|\boldsymbol{x}^j| = \sum_{i=1}^{V_j}x_i^j$ as $\tilde{\boldsymbol{x}}^j = \boldsymbol{x}^j/|\boldsymbol{x}^j|$, we obtain the class posterior distribution for Hybrid:

$$R(k|\tilde{\boldsymbol{x}};\Theta,\Lambda) = \frac{e^{\mu_k}\prod_{j=1}^{J}\prod_{i=1}^{V_j}(\theta_{ki}^j)^{\lambda_j\tilde{x}_i^j}}{\sum_{k'=1}^{K}e^{\mu_{k'}}\prod_{j=1}^{J}\prod_{i=1}^{V_j}(\theta_{k'i}^j)^{\lambda_j\tilde{x}_i^j}}, \quad \forall k. \tag{17}$$

The class posterior distribution for Hybrid-L is expressed by using $\lambda_{jk}$ instead of $\lambda_j$ in Eq. (17).

For MAP estimation of NB parameter $\boldsymbol{\theta}_k^j$, as the prior $p(\boldsymbol{\theta}_k^j)$ in Eq. (2), we use a Dirichlet prior: $p(\boldsymbol{\theta}_k^j) \propto \prod_{i=1}^{V_j}(\theta_{ki}^j)^{\xi_k^j-1}$, where $\xi_k^j(>1)$ represents a hyperparameter. Let $\{\tilde{\boldsymbol{x}}_m\}_{m=1}^{M_k}$ represent the normalized feature vectors of training samples that belong to the $k$th class. Then, the estimate of $\theta_{ki}^j$ is computed as

$$\hat{\theta}_{ki}^j = \frac{\sum_{m=1}^{M_k}\tilde{x}_{mi}^j + \xi_k^j - 1}{M_k + V_j(\xi_k^j - 1)}. \tag{18}$$

For Hybrid and Hybrid-L, we tune the hyperparameter $\xi_k^j$ to maximize the sum of the log likelihood computed with a leave-one-out cross-validation of the training samples,

$$L(\xi_k^j) = \sum_{m=1}^{M_k} \log P(\tilde{\boldsymbol{x}}_m^j | k; \hat{\boldsymbol{\theta}}_k^{j,(-m)}) = \sum_{m=1}^{M_k} \sum_{i=1}^{V_j} \tilde{x}_{mi}^j \log \hat{\theta}_{ki}^{j,(-m)}, \tag{19}$$

because we confirmed this tuning was practically useful for classification. Here, $\hat{\boldsymbol{\theta}}_k^{j,(-m)} = \{\hat{\theta}_{ki}^{j,(-m)}\}_i$ is the estimate of $\boldsymbol{\theta}_k^j = \{\theta_{ki}^j\}_i$ computed by using training samples other than $\tilde{\boldsymbol{x}}_m$. This tuning is executed with the help of the EM algorithm (Dempster, Laird, & Rubin, 1977). See Appendix A for the details.

## 4. Experiments

### 4.1. Test collections

An empirical evaluation was performed on four test collections: *20 newsgroups (20news), NIPS*,[1] *WebKB*, and *Cora*[2] datasets. 20news, WebKB (Nigam et al., 1999), and Cora have often been used as benchmark tests of classifiers in text classification tasks, and NIPS is the ASCII text collection of papers from NIPS conferences created by Yann using optical character recognition.

20news consists of 20 different UseNet discussion groups and contains 18,828 articles. Each article belongs to one of the 20 groups. We extracted two components, *Main* (M) and *Title* (T), from each article, where T is the text description following "Subject:" and M is the main information in each article except for the title. Each component contains words as features. We removed vocabulary words included either in the stoplist (Salton & McGill, 1983) or in only one article. There were 52,313 and 5320 vocabulary words, respectively, in components M and T in the dataset.

NIPS consists of 1740 papers from NIPS conferences 1–12. We used 1164 papers from conferences 5–12 in our experiments. Each paper is related to one of nine research topics, for example, neuroscience, theory, and applications. We extracted four components, *Main* (M), *Title* (T), *Abstract* (A), and *References* (R), from each paper, where M is the main information in each paper excluding the title, abstract, and references. We removed vocabulary words in the same way as for 20 news. There were 20,485, 904, 5021, and 8303 vocabulary words, respectively, in components M, T, A, and R in the dataset.

WebKB contains Web pages from universities. This dataset consists of seven categories, and each page belongs to one of these categories. Following the setup in (Nigam et al., 1999), only four categories *course, faculty, project*, and *student* were used. The categories contained a total of 4199 pages. We extracted six components, *Main* (M), *Title* (T), *In-Links* (IL), *Out-Links* (OL), *File-Links* (FL), and *Anchor-Text* (AT), from each page. Here, T is the text description between ⟨TITLE⟩ and ⟨/TITLE⟩ tags, and M is the main information except for the title, tags, and links. IL consists of links from other pages. AT is the set of anchor text for each page, which consists of text descriptions that express the links to the page found on the other pages. We collected IL and AT from the links within the dataset. OL consists of links to other pages, and FL consists of links to files such as images. M, T, and AT contain words as features, and IL, OL, and FL contain URLs of Web pages or files. We removed vocabulary words in the same way as for 20news and removed URLs included in only one page for each component. There were 18,471, 995, and 496 vocabulary words, respectively, in components M, T, and AT in the dataset. Components IL, OL, and FL contained 500, 4131, and 484 different URLs, respectively.

Cora contains more than 30,000 summaries of technical papers that belong to one of 70 groups. We extracted five components, *Abstract* (A), *Title* (T), *AUthors* (AU), *In-Links* (IL), and *Out-Links* (OL) from each paper. Here, T and A are the text distribution included in the papers, and AU is the set of authors. IL consists of links (citations) from other papers, and OL consists of links (citations) to other papers. IL and OL contain paper ID numbers as features. For our evaluation, we used 10,782 papers included in 18

[1] http://www.cs.toronto.edu/~roweis/data.html
[2] http://www.cs.umass.edu/~mccallum/data/cora-classify.tar.gz

groups: */Artificial_Intelligence/**. We removed vocabulary words in the same way as for 20news and removed authors and paper ID numbers included in only one paper for each component. There were 15,040, 4209, 3813, 6773, and 33,569 vocabulary words (authors/paper ID numbers), respectively, in components A, T, AU, IL, and OL in the dataset.

Table 1 shows the properties of the components in the four datasets. $|D_c|/|D_t|$ for each component in Table 1 shows the percentage of data samples whose components are not empty. In 20news, NIPS, and WebKB, $|D_c|/|D_t|$ for M was close to 100%. On the other hand, $|D_c|/|D_t|$ for the other components was small, especially for AT, IL, OL, and FL with respect to hyperlinks in WebKB and IL in Cora. $|F|/|D_t|$ for each component in Table 1 shows the average number of features contained by the component and implies the component size. In 20news, NIPS, and WebKB, $|F|/|D_t|$ for M was much larger than for other components.

### 4.2. Experimental settings

#### 4.2.1. Evaluation methods

To evaluate our hybrid classifiers, Hybrid and Hybrid-L, we compared their classification performance with that of other classifiers. In *Experiment 1*, we compared Hybrid and Hybrid-L with generative and discriminative classifiers designed solely by using a main component, to examine the effect of additional components on classification performance. In *Experiment 2*, we compared Hybrid with pure generative and discriminative classifiers designed to deal with all components as presented in Section 2, to confirm the effect of our hybrid approach on classification performance. In *Experiment 3*, we compared Hybrid with the weighted average and product of individual classifiers designed by single components (component classifiers), which have often been used for dealing with multiple components.

Table 1
The properties of components in datasets

| Component | M | T |
|---|---|---|
| *(a) 20news ($\|D_t\|$ = 18,828)* | | |
| $\|D_c\|$ | 18,782 | 18,456 |
| $\|D_c\|/\|D_t\|$ | 99.8 % | 98.0 % |
| $\|F\|$ | 1,960,166 | 59,261 |
| $\|F\|/\|D_t\|$ | 104.1 | 3.1 |

| | M | T | A | R |
|---|---|---|---|---|
| *(b) NIPS ($\|D_t\|$ = 1164)* | | | | |
| $\|D_c\|$ | 1164 | 1160 | 1164 | 1164 |
| $\|D_c\|/\|D_t\|$ | 100% | 99.7% | 100% | 100% |
| $\|F\|$ | 1,386,484 | 5903 | 74,385 | 161,992 |
| $\|F\|/\|D_t\|$ | 1191.1 | 5.1 | 63.9 | 139.2 |

| | M | T | AT | IL | OL | FL |
|---|---|---|---|---|---|---|
| *(c) WebKB ($\|D_t\|$ = 4199)* | | | | | | |
| $\|D_c\|$ | 4199 | 3851 | 1101 | 1242 | 3273 | 969 |
| $\|D_c\|/\|D_t\|$ | 100% | 91.7% | 26.2% | 29.6% | 77.9% | 23.1% |
| $\|F\|$ | 668,192 | 10,403 | 5882 | 2547 | 16,535 | 3165 |
| $\|F\|/\|D_t\|$ | 159.1 | 2.5 | 1.4 | 0.6 | 3.9 | 0.8 |

| | A | T | AU | IL | OL |
|---|---|---|---|---|---|
| *(d) Cora ($\|D_t\|$ = 10,782)* | | | | | |
| $\|D_c\|$ | 9217 | 10274 | 9343 | 5265 | 10617 |
| $\|D_c\|/\|D_t\|$ | 85.5% | 96.2% | 86.7% | 48.8% | 98.5% |
| $\|F\|$ | 670,839 | 61,003 | 19,331 | 30,577 | 175,248 |
| $\|F\|/\|D_t\|$ | 62.2 | 5.7 | 1.8 | 2.8 | 16.3 |

For each component, $|D_c|$ is the number of data samples whose components contain features, and $|F|$ is the number of features contained over all the data samples. $|D_t|$ is the total number of data samples in each dataset.

#### 4.2.2. Evaluation measure

We examined classification accuracies with test samples to compare Hybrid with other classifiers. In our experiments, we selected the training and test samples randomly from each dataset. We made 10 different evaluation sets for each dataset by random selection. For 20news, NIPS, WebKB, and Cora, respectively, we selected 8000, 500, 2000, 5000 data samples as test samples for each evaluation set. After extracting the test samples, training samples were selected from the remaining data samples in each dataset. The average classification accuracy over the 10 evaluation sets was used to evaluate the classifiers with each dataset.

### 4.3. Experiment 1

#### 4.3.1. Compared classifiers

To confirm the effect of additional components on classification performance, we examined the classification accuracies with NB and MLR classifiers constructed solely using a *main* component and compared them with Hybrid and Hybrid-L. To construct the NB and NLR classifiers, component M was used for 20news, NIPS, and WebKB. For Cora, component OL was used because it provided the best classification performance of the five components.

#### 4.3.2. Results

Table 2 shows the average classification accuracies over the 10 different evaluation sets for (a) 20news, (b) NIPS, (c) WebKB, and (d) Cora. Each number in parentheses in the table denotes the standard deviation of the 10 evaluation sets. $|D|$ represents the number of training samples. An asterisk in the column of each

Table 2
Classification accuracies (%) with Hybrid, Hybrid-L, NB with a main component, and MLR with a main component

| $\|D\|$ | Hybrid | Hybrid-L | NB | MLR |
|---|---|---|---|---|
| *(a) 20news* | | | | |
| 160 | **50.8** (2.0) | 46.5 (1.8)* | 45.1 (1.6)* | 45.0 (1.8)* |
| 320 | **61.4** (1.4) | 59.4 (1.2)* | 54.0 (1.2)* | 54.7 (1.6)* |
| 640 | **70.9** (0.9) | 70.1 (1.0)* | 62.6 (1.0)* | 63.7 (1.1)* |
| 1280 | **78.6** (0.5) | 78.2 (0.4)* | 69.4 (0.9)* | 70.7 (0.5)* |
| 2560 | **84.3** (0.5) | 84.1 (0.5)* | 75.8 (0.6)* | 76.1 (0.5)* |
| 5120 | **88.5** (0.2) | 88.4 (0.3)* | 81.2 (0.5)* | 80.5 (0.5)* |
| 10240 | **91.5** (0.2) | **91.5** (0.2) | 85.4 (0.4)* | 84.4 (0.3)* |
| *(b) NIPS* | | | | |
| 72 | **60.0** (2.8) | 56.6 (2.7)* | 57.5 (2.8) | 51.7 (3.2)* |
| 144 | **65.1** (1.6) | 63.3 (1.9)* | 61.5 (2.2)* | 57.8 (2.1)* |
| 288 | **69.5** (1.8) | 67.2 (1.5)* | 67.7 (1.5)* | 62.0 (2.0)* |
| 576 | **72.0** (1.9) | 71.2 (1.6)* | 70.8 (1.9)* | 66.3 (1.9)* |
| *(c) WebKB* | | | | |
| 32 | **65.3** (2.9) | 51.9 (1.5)* | 63.5 (4.4) | 60.5 (5.8)* |
| 64 | **74.9** (2.7) | 67.2 (3.3)* | 71.3 (3.9)* | 70.0 (3.7)* |
| 128 | **82.1** (1.6) | 78.3 (2.2)* | 77.1 (2.4)* | 78.2 (2.4)* |
| 256 | **86.8** (1.1) | 84.3 (1.0)* | 80.1 (1.5)* | 83.5 (1.3)* |
| 512 | **88.9** (0.7) | 87.9 (0.7)* | 80.8 (3.1)* | 87.4 (0.9)* |
| 1024 | **90.8** (0.7) | 90.5 (0.8) | 82.7 (1.8)* | 89.7 (0.5)* |
| 2048 | 92.1 (0.4) | **92.2** (0.3)* | 82.8 (1.0)* | 91.0 (0.7)* |
| *(d) Cora* | | | | |
| 144 | **51.2** (1.9) | 41.0 (2.3)* | 43.7 (1.6)* | 40.9 (1.4)* |
| 288 | **61.7** (1.5) | 55.5 (1.6)* | 54.1 (0.8)* | 52.0 (1.0)* |
| 576 | **70.2** (0.7) | 67.5 (0.6)* | 63.7 (0.5)* | 62.2 (0.7)* |
| 1152 | **76.2** (0.5) | 75.0 (0.6)* | 71.0 (0.8)* | 69.3 (0.7)* |
| 2304 | **80.2** (0.4) | 79.8 (0.4)* | 76.1 (0.6)* | 74.7 (0.6)* |
| 4608 | **83.3** (0.3) | 83.1 (0.3)* | 80.4 (0.6)* | 78.5 (0.6)* |

compared classifier shows that the difference between the average classification accuracies of the classifier and Hybrid is significant ($p < 0.05$) in the Wilcoxon test.

In all cases, Hybrid achieved higher average classification accuracy than the NB and MLR classifiers only using a main component. The difference between the average classification accuracies of Hybrid and the NB classifier was significant in the Wilcoxon test, except when $|D| = 72$ for NIPS and $|D| = 32$ for WebKB. The difference between Hybrid and the MLR classifier was significant in all cases. Additional components contributed toward improving the classification performance.

However, Hybrid-L did not always outperform the NB and MLR classifiers, when the number of training samples was small. With a large number of training samples, the classification performance of Hybrid-L was better than the that of NB and MLR classifiers and similar to that of Hybrid. This indicates that Hybrid-L is more overfitted with training samples than Hybrid.

### 4.4. Experiment 2

#### 4.4.1. Compared classifiers

To confirm the effect of our hybrid approach on classification performance, we compared Hybrid with NB and MLR based classifiers designed to deal with *all* components included in data samples. As one NB (MLR) based classifier, we employed a *product*-based NB (MLR) classifier "PNB (PMLR)" designed based on the *simple product* of component NB (MLR) models as presented in Section 2. We also employed a *single*-model based NB (MLR) classifier "SNB (SMLR)" designed by using a single NB (MLR) model that deals with all components. Although the single NB model might be inappropriate when data samples consist of different types of media, we examined the SNB classification performance to evaluate Hybrid.

#### 4.4.2. Results

Table 3 shows the average classification accuracies obtained with Hybrid, PNB, PMLR, SNB, and SMLR. An asterisk in the column of each compared classifier shows that the difference between the average classification accuracies of the classifier and Hybrid is significant ($p < 0.05$) in the Wilcoxon test.

For all datasets, Hybrid outperformed PNB. This result indicates that the combination weights of component generative models provided by the ME principle is effective in improving the classification performance.

Hybrid provided the best performance of the five except when $|D| = 2048$ for WebKB. When the pure generative classifiers, PNB and SNB, and the pure discriminative classifiers, PMLR and SMLR, performed similarly, Hybrid performed much better than these pure classifiers. We confirmed that our hybrid approach was effective in improving the classification performance especially when the classification performance of the pure generative approaches was similar to that of the pure discriminative approaches.

#### 4.4.3. Analysis of combination weights

We examined the combination weights of component generative models that were estimated in our hybrid approach. Each circle in Fig. 2 represents the average estimate of combination weight $\lambda_j$ in Hybrid, which was trained by using 10,240, 576, 2048, and 4608 training samples for 20news, NIPS, WebKB, and Cora, respectively. Each bar in Fig. 2 represents the average classification accuracy obtained with an NB classifier designed using a single component (NB component classifier). The average classification accuracy of the NB component classifier was examined using only test samples whose components were not empty. Each triangle in Fig. 2 indicates $\alpha_j = \lambda_M |F|_j / |F|_M$, where $|F|_M$ and $\lambda_M$ represent $|F|$ shown in Table 1 and $\lambda_j$ for component M. When the number of features contained by the $j$th component, $|F|_j$, is used as a measure of the component size, $\alpha_j$ means the ratio of the component size.

As shown in Fig. 2, estimates of combination weights in Hybrid tended to be large for the components that obtained high average classification accuracies. For WebKB, the average estimate of $\lambda_j$ for IL was larger than for T and OL, and the performance of the NB component classifier provided by IL was better, although the component size of IL was smaller. This indicates that our hybrid approach provides the combination weights from the classification performance of components rather than from the component size. We confirmed that Hybrid used *minor* components effectively by providing large combination weights for the components whose sizes were small but where the classification performance was good.

Table 3
Classification accuracies (%) with Hybrid, PNB, PMLR, SNB, and SMLR

| $\|D\|$ | Hybrid | PNB | PMLR | SNB | SMLR |
|---|---|---|---|---|---|
| *(a) 20news* | | | | | |
| 160 | **50.8** (2.0) | 47.4 (1.8)* | 47.8 (1.4)* | 47.5 (1.8)* | 46.7 (1.8)* |
| 320 | **61.4** (1.4) | 56.9 (1.3)* | 57.4 (1.6)* | 56.9 (1.3)* | 56.8 (1.8)* |
| 640 | **70.9** (0.9) | 66.0 (1.2)* | 66.6 (0.7)* | 66.0 (1.1)* | 65.9 (1.1)* |
| 1280 | **78.6** (0.5) | 73.3 (0.8)* | 74.7 (0.4)* | 73.3 (0.9)* | 73.3 (0.5)* |
| 2560 | **84.3** (0.5) | 79.8 (0.6)* | 80.9 (0.4)* | 79.8 (0.6)* | 78.8 (0.6)* |
| 5120 | **88.5** (0.2) | 85.0 (0.4)* | 85.6 (0.3)* | 85.0 (0.4)* | 83.3 (0.4)* |
| 10,240 | **91.5** (0.2) | 88.9 (0.3)* | 89.5 (0.2)* | 88.9 (0.3)* | 87.3 (0.2)* |
| *(b) NIPS* | | | | | |
| 72 | **60.0** (2.8) | 58.4 (2.8) | 51.5 (2.4)* | 57.7 (2.5) | 51.5 (2.9)* |
| 144 | **65.1** (1.6) | 63.0 (1.8)* | 57.5 (2.4)* | 62.6 (2.2)* | 58.2 (2.5)* |
| 288 | **69.5** (1.8) | 68.5 (1.6) | 62.6 (1.8)* | 68.6 (1.5)* | 62.6 (1.9)* |
| 576 | **72.0** (1.9) | 71.5 (1.7) | 65.8 (1.6)* | 71.6 (1.6) | 66.6 (1.9)* |
| *(c) WebKB* | | | | | |
| 32 | **65.3** (2.9) | 64.5 (4.6) | 61.7 (6.1) | 63.5 (4.6) | 60.4 (5.9)* |
| 64 | **74.9** (2.7) | 72.4 (4.2) | 70.9 (3.3)* | 71.8 (4.5)* | 69.9 (4.2)* |
| 128 | **82.1** (1.6) | 78.7 (2.4)* | 78.1 (1.3)* | 78.3 (2.6)* | 78.5 (2.3)* |
| 256 | **86.8** (1.1) | 82.0 (1.6)* | 82.6 (0.4)* | 81.9 (1.7)* | 84.2 (1.3)* |
| 512 | **88.9** (0.7) | 83.6 (2.8)* | 85.8 (0.8)* | 83.6 (2.7)* | 88.1 (0.9)* |
| 1024 | **90.8** (0.7) | 85.7 (1.6)* | 88.1 (0.6)* | 85.8 (1.5)* | **90.8** (0.5) |
| 2048 | 92.1 (0.4) | 86.6 (0.8)* | 89.2 (0.4)* | 86.5 (0.8)* | **92.6** (0.4)* |
| *(d) Cora* | | | | | |
| 144 | **51.2** (1.9) | 45.8 (2.6)* | 47.0 (2.1)* | 42.2 (4.2)* | 44.3 (2.5)* |
| 288 | **61.7** (1.5) | 57.0 (1.9)* | 57.3 (1.5)* | 52.2 (3.3)* | 54.3 (2.0)* |
| 576 | **70.2** (0.7) | 66.8 (0.9)* | 66.1 (0.8)* | 62.7 (1.5)* | 63.3 (1.3)* |
| 1152 | **76.2** (0.5) | 73.9 (0.5)* | 72.2 (0.5)* | 71.6 (0.5)* | 70.0 (0.5)* |
| 2304 | **80.2** (0.4) | 78.7 (0.4)* | 76.7 (0.3)* | 77.8 (0.4)* | 75.0 (0.4)* |
| 4608 | **83.3** (0.3) | 82.2 (0.4)* | 79.8 (0.4)* | 82.0 (0.5)* | 78.6 (0.4)* |

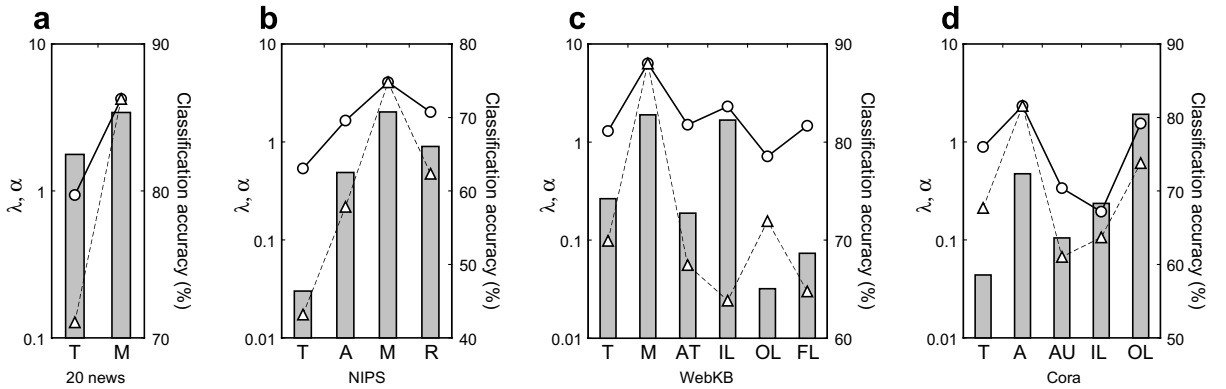


Fig. 2. Combination weights of component models in Hybrid, $\lambda_j$ (circles), ratio of component size, $\alpha_j$ (triangles), and classification accuracies with NB component classifiers (bars).

As reported in (Lee, 1995), it is known that combining different representations of data samples is often more effective for improving performance in information retrieval tasks than combining similar ones. Our experimental results also showed that Hybrid was constructed mainly by combining different representations. For WebKB (Cora), the estimates of combination weights for text component M (A) and link component IL (OL) were larger than for other components. We can suppose that our hybrid approach automatically

provides combination weights and thus effectively uses different representations to improve classification performance.

### 4.5. Experiment 3

#### 4.5.1. Compared classifiers

Our experimental results in Section 4.4.3 suggest that it is promising to combine components with weights induced from the classification performance of individual components. In Hybrid, such weights are estimated with *little exploration*, as the global maximum point of the upper convex objective function shown in Eq. (12). However, we can also consider the direct use of the classification accuracies of individual component classifiers to obtain their combination weights with little exploration. Thus, Hybrid was compared with classifiers based on the weighted combination of component NB (MLR) classifiers. As simple formulations for the weighted combination, we considered the weighted average $\sum_{j=1}^{J}\gamma_j P(k|\boldsymbol{x}^j)$ and product $\prod_{j=1}^{J}P(k|\boldsymbol{x}^j)^{\gamma_j}$ of component classifiers $\{P(k|\boldsymbol{x}^j)\}_j$, where the ratio of classification accuracies of $\{P(k|\boldsymbol{x}^j)\}_j$ was applied to $\Gamma=\{\gamma_j\}_j$. We call the weighted average of component NB (MLR) classifiers "WANB (WAMLR)" and the weighted product of component NB (MLR) classifiers "WPNB (WPMLR)".

#### 4.5.2. Results

Table 4 shows the average classification accuracies obtained with Hybrid, WANB, WAMLR, WPNB, and WPMLR. An asterisk in the column of each compared classifier shows that the difference between the average classification accuracies of the classifier and Hybrid is significant ($p < 0.05$) in the Wilcoxon test.

Table 4
Classification accuracies (%) with Hybrid, WANB, WAMLR, WPNB, and WPMLR

| $\|D\|$ | Hybrid | WANB | WAMLR | WPNB | WPMLR |
|---|---|---|---|---|---|
| *(a) 20news* | | | | | |
| 160 | **50.8** (2.0) | 45.8 (1.8)* | 48.2 (1.6)* | 46.6 (1.8)* | 48.7 (1.6)* |
| 320 | **61.4** (1.4) | 55.0 (1.3)* | 58.0 (1.7)* | 56.2 (1.3)* | 58.8 (1.7)* |
| 640 | **70.9** (0.9) | 63.7 (1.0)* | 67.3 (0.7)* | 65.3 (1.1)* | 67.9 (0.7)* |
| 1280 | **78.6** (0.5) | 71.0 (0.8)* | 75.1 (0.5)* | 72.8 (0.8)* | 75.5 (0.4)* |
| 2560 | **84.3** (0.5) | 77.7 (0.6)* | 81.2 (0.3)* | 79.6 (0.6)* | 81.3 (0.3)* |
| 5120 | **88.5** (0.2) | 83.3 (0.5)* | 85.8 (0.3)* | 84.9 (0.4)* | 85.8 (0.3)* |
| 10,240 | **91.5** (0.2) | 87.6 (0.3)* | 89.6 (0.2)* | 88.8 (0.3)* | 89.5 (0.2)* |
| *(b) NIPS* | | | | | |
| 72 | **60.0** (2.8) | 58.9 (2.2) | 53.5 (2.9)* | 58.5 (2.8) | 53.0 (3.4)* |
| 144 | **65.1** (1.6) | 63.4 (2.1)* | 59.9 (1.7)* | 63.0 (1.8)* | 58.8 (2.4)* |
| 288 | **69.5** (1.8) | 67.7 (1.9)* | 64.2 (1.4)* | 68.4 (1.5) | 63.6 (1.8)* |
| 576 | **72.0** (1.9) | 70.6 (1.7)* | 68.3 (2.2)* | 71.5 (1.7) | 67.5 (1.8)* |
| *(c) WebKB* | | | | | |
| 32 | **65.3** (2.9) | 64.7 (5.2) | 63.9 (5.5) | 65.0 (4.2) | 62.5 (5.8) |
| 64 | **74.9** (2.7) | 73.4 (5.5) | 72.8 (3.9) | 73.2 (3.8) | 71.9 (3.8)* |
| 128 | **82.1** (1.6) | 78.0 (5.1)* | 81.0 (1.3) | 78.8 (2.2)* | 79.6 (1.5)* |
| 256 | **86.8** (1.1) | 79.6 (4.7)* | 85.7 (0.8)* | 82.2 (1.3)* | 84.4 (0.7)* |
| 512 | 88.9 (0.7) | 79.2 (4.2)* | **89.0** (1.0) | 83.6 (3.0)* | 87.4 (1.0)* |
| 1024 | 90.8 (0.7) | 81.7 (1.7)* | **91.0** (0.5) | 85.9 (1.6)* | 89.5 (0.5)* |
| 2048 | **92.1** (0.4) | 82.6 (1.1)* | 91.9 (0.4) | 86.7 (0.8)* | 90.5 (0.4)* |
| *(d) Cora* | | | | | |
| 144 | **51.2** (1.9) | 46.0 (2.5)* | 47.4 (2.0)* | 43.3 (2.5)* | 48.6 (1.9)* |
| 288 | **61.7** (1.5) | 57.6 (2.1)* | 58.4 (1.5)* | 54.7 (2.1)* | 59.2 (1.6)* |
| 576 | **70.2** (0.7) | 67.3 (0.8)* | 67.8 (1.0)* | 64.7 (0.7)* | 68.2 (0.8)* |
| 1152 | **76.2** (0.5) | 73.7 (0.8)* | 74.3 (0.5)* | 72.0 (0.6)* | 73.9 (0.4)* |
| 2304 | **80.2** (0.4) | 77.9 (0.7)* | 78.9 (0.3)* | 77.0 (0.6)* | 77.9 (0.2)* |
| 4608 | **83.3** (0.3) | 81.0 (0.6)* | 82.1 (0.4)* | 80.7 (0.6)* | 80.7 (0.3)* |

As shown in the table, for 20news, NIPS, and Cora, the classification performance of Hybrid was the best of the five. For WebKB, Hybrid provided better or similar performance to the other classifiers. We confirmed experimentally that the combination of components in our hybrid approach was more effective for improving classification performance than the weighted average and product of component classifiers with the ratio of their classification accuracies.

## 5. Related work

We presented a text classifier constructed by individually modeling multiple components included in text data samples and discriminatively combining these models according to the ME principle. In the fields of machine translation and speech recognition, classifiers have been used that were designed by combining multiple probabilistic models on the basis of discriminative approaches. For machine translation, a combination of translation and linguistic models based on the ME principle was proposed in (Och & Ney, 2002). For speech recognition, discriminative combinations of various models for acoustic, linguistic, and visual information were proposed in (Beyerlein, 1998; Glotin, Vergyri, Neti, Potamianos, & Luettin, 2001).

## 6. Conclusion

We proposed a new classifier that uses both main text and additional components effectively for multiclass and single-labeled text classification problems based on a hybrid consisting of generative and discriminative approaches. The main idea is to design an individual component generative model for each component and combine all of these models according to the maximum entropy (ME) principle, where the same combination weight of a component is provided for all classes. We also considered another hybrid classifier, Hybrid-L, obtained by the ME principle, where a combination weight of a component is provided per class.

In our experiments using four datasets, the classification performance of our hybrid classifier was better than or similar to that of Hybrid-L. Next, we confirmed that our hybrid classifier often outperformed pure generative and discriminative classifiers. Our hybrid classifier was useful especially when the classification performance of the pure generative classifiers was comparable to that of pure discriminative classifiers. We also confirmed that our hybrid classifier performed better than classifiers constructed by combining component classifiers with weights, as determined by the classification performance of the component classifiers. We believe that the hybrid approach improved the classification performance by providing combination weights of component generative models on the basis of the discriminative approach.

Future work will involve applying our hybrid classifier to multimodal data in which different generative models are employed, to confirm that the hybrid approach is useful for dealing with text and various additional components. We will try to train the hybrid classifier with labeled and unlabeled samples, which are data samples with and without class labels.

## Appendix A. Hyperparameter tuning procedure

We explain the procedure for tuning hyberparameter $\xi_k^j$ using a leave-one-out cross-validation and the EM algorithm as mentioned in Section 3.2. According to MAP estimation using training samples except $\tilde{\boldsymbol{x}}_m$, we obtain $\hat{\theta}_{ki}^{j,(-m)}$ in Eq. (19), as

$$\hat{\theta}_{ki}^{j,(-m)} = \frac{\sum_{m'=1}^{M_k} \tilde{x}_{m'i}^j - \tilde{x}_{mi}^j + \xi_k^j - 1}{M_k - 1 + V_j(\xi_k^j - 1)}. \tag{A.1}$$

As with parameter estimates smoothed by Lidstone's law (cf. Manning & Schütze, 1999), we can express $\hat{\theta}_{ki}^{j,(-m)}$ by

$$\hat{\theta}_{ki}^{j,(-m)} = \beta\psi_i^{(-m)} + (1-\beta)\frac{1}{V_j}, \tag{A.2}$$

where

$$\beta = \frac{M_k - 1}{M_k - 1 + V_j(\xi_k^j - 1)}, \quad 0 \leqslant \beta < 1, \tag{A.3}$$

$$\psi_i^{(-m)} = \frac{\sum_{m'=1}^{M_k} \tilde{x}_{m'i}^j - \tilde{x}_{mi}^j}{M_k - 1} \geqslant 0, \quad \sum_{i=1}^{V_j} \psi_i^{(-m)} = 1. \tag{A.4}$$

Therefore, we can view $\hat{\theta}_{ki}^{j,(-m)}$ as a linear interpolation between $\psi_i^{(-m)}$ and $1/V_j$. Since $\beta$ is independent of the training sample $\tilde{\boldsymbol{x}}_m$, we can regard $L(\xi_k^j)$ shown in Eq. (19) as a function of $\beta$:

$$L(\beta) = \sum_{m=1}^{M_k} \sum_{i=1}^{V_j} \tilde{x}_{mi}^j \log \left\{ \beta \psi_i^{(-m)} + (1 - \beta) \frac{1}{V_j} \right\}. \tag{A.5}$$

We can use the EM algorithm for estimating $\beta$ to maximize $L(\beta)$. In this estimation, global optimality is guaranteed, since $L(\beta)$ is an upper convex function. Such an estimation of interpolation weight $\beta$ with cross-validation was also applied in *deleted interpolation* (Jelinek & Mercer, 1980). Using the estimate of $\beta$ and Eq. (A.3), we obtain $\xi_k^j$ to maximize $L(\xi_k^j)$ shown in Eq. (19).

## References


Berger, A. L., Della Pietra, S. A., & Della Pietra, V. J. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics, 22*(1), 39–71.

Beyerlein, P. (1998). Discriminative model combination. In *Proceedings of the IEEE international conference on acoustics, speech, and signal processing (ICASSP1998)*, pp. 481–484.

Brochu, E., & Freitas, N. (2003). "Name that song!": A probabilistic approach to querying on music and text. *Advances in neural information processing systems* (Vol. 15, pp. 1505–1512). Cambridge, MA: MIT Press.

Chakrabarti, S., Dom, B., & Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. In *Proceedings of ACM international conference on management of data (SIGMOD-98)*, pp. 307–318.

Chen, S. F., & Rosenfeld, R. (1999). A Gaussian prior for smoothing maximum entropy models. Technical report, Carnegie Mellon University.

Cohn, D., & Hofmann, T. (2001). The missing link – a probabilistic model of document content and hypertext connectivity. *Advances in neural information processing systems* (Vol. 13, pp. 430–436). Cambridge, MA: MIT Press.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B, 39*, 1–38.

Glotin, H., Vergyri, D., Neti, C., Potamianos, G., & Luettin, J. (2001). Weighting schemes for audio-visual fusion in speech recognition. In *Proceedings of the IEEE international conference on acoustics, speech, and signal processing (ICASSP2001)*, pp. 165–168.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction*. New York, Berlin, Heidelberg: Springer-Verlag.

Jelinek, F., & Mercer, R. (1980). Interpolated estimation of Markov source parameters from sparse data. In E. S. Gelsema & L. N. Kanal (Eds.), *Pattern recognition in practice* (pp. 381–402). Amsterdam, The Netherlands: North-Holland Publishing Company.

Lee, J. H. (1995). Combining multiple evidence from different properties of weighting schemes. In *Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'95)*, pp. 180–188.

Liu, D. C., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming, Series B, 45*(3), 503–528.

Lu, Q., & Getoor, L. (2003). Link-based text classification. In *IJCAI workshop on text-mining & link-analysis* (TextLink 2003).

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, Massachusetts: The MIT Press.

Nigam, K., Lafferty, J., & McCallum, A. (1999). Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, pp. 61–67.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*, 103–134.

Och, F. J., & Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th annual meeting of the association for computational linguistics (ACL2002)*, pp. 295–302.

Raina, R., Shen, Y., Ng, A. Y., & McCallum, A. (2004). Classification with hybrid generative/discriminative models. *Advances in Neural Information Processing Systems* (Vol. 16). Cambridge, MA: MIT Press.

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. New York: McGraw-Hill.

Sun, A., Lim, E. P., & Ng, W. K. (2002). Web classification using support vector machine. In *Proceedings of 4th international workshop on Web information and data management (WIDM 2002) held in conj. with CIKM 2002*, pp. 96–99.