Semisupervised Learning for a Hybrid Generative/Discriminative Classifier Based on the Maximum Entropy Principle

Akinori Fujino, *Member*, *IEEE*, Naonori Ueda, *Member*, *IEEE*, and Kazumi Saito

Abstract—This paper presents a method for designing semisupervised classifiers trained on labeled and unlabeled samples. We focus on a probabilistic semisupervised classifier design for multiclass and single-labeled classification problems and propose a hybrid approach that takes advantage of generative and discriminative approaches. In our approach, we first consider a generative model trained by using labeled samples and introduce a bias correction model, where these models belong to the same model family but have different parameters. Then, we construct a hybrid classifier by combining these models based on the maximum entropy principle. To enable us to apply our hybrid approach to text classification problems, we employed naive Bayes models as the generative and bias correction models. Our experimental results for four text data sets confirmed that the generalization ability of our hybrid classifier was much improved by using a large number of unlabeled samples for training when there were too few labeled samples to obtain good performance. We also confirmed that our hybrid approach significantly outperformed the generative and discriminative approaches when the performance of the generative and discriminative approaches was comparable. Moreover, we examined the performance of our hybrid classifier when the labeled and unlabeled data distributions were different.

Index Terms—Generative model, maximum entropy principle, bias correction, unlabeled samples, text classification.

1 Introduction

Statistical classifiers are generally trained by using observed feature vectors with class labels, called *labeled* samples. If we are to obtain better classifiers with a generalization ability, then we require a large number of labeled samples. However, in practice, it is often fairly expensive to collect many labeled samples because class labels are manually assigned by experienced analysts. In contrast, *unlabeled* samples can be easily collected. Therefore, effectively utilizing unlabeled samples to improve the generalization performance of classifiers is a major research issue in the fields of pattern recognition and machine learning, and semisupervised learning methods that use both labeled and unlabeled samples for training classifiers have been developed [1], [2], [3], [4], [5], [6], [7], [8] (see [9] for a comprehensive survey). In this paper, we focus on designing semisupervised classifiers for multiclass and single-labeled classification based on probabilistic approaches.

Semisupervised learning algorithms based on probabilistic approaches have been proposed for *generative* and *discriminative* classifiers. Generative classifiers learn the joint probability model $p(x,y)$ of the feature vector x and class label y of a data sample and make their predictions by using Bayes rule to compute $P(y|x)$ and then taking the most probable label y. For semisupervised learning of the classifier, unlabeled samples are dealt with as a missing class label problem and are incorporated in a mixture of joint probability models [1]. One of the authors has presented an algorithm for incorporating unlabeled sequential data with a mixture of hidden Markov models and confirmed experimentally that the algorithm was useful for improving their classification performance [4].

By contrast, discriminative classifiers model class posterior probability $P(y|x)$ and learn mapping from x to y directly. Since $p(x)$ is not modeled in the discriminative approach, some assumptions are required if we are to incorporate unlabeled samples in the model. In [3], it is assumed that if two feature vectors are close, then the class labels of both samples should be the same. The *minimum entropy regularizer (MER)* was recently introduced as another approach to semisupervised learning [2]. In [2], by utilizing the knowledge that unlabeled samples are beneficial for improving classification accuracy when samples are well separated among classes, an attempt was made to minimize the entropy of the class posterior probabilities of unlabeled samples.

Semisupervised learning algorithms are desired when there are insufficient labeled samples to obtain good supervised classifiers with a generalization ability. In supervised learning cases, it has been shown that discriminative classifiers often achieve better performance than generative classifiers but that generative classifiers often provide better generalization performance than discriminative classifiers when trained with few labeled samples [10]. Therefore, we explore a *hybrid* of generative and discriminative approaches to benefit from their respective advantages and, thus, obtain semisupervised classifiers with good performance.

Supervised classifiers based on the *hybrid* generative and discriminative approaches have recently been proposed [11], [12]. In [11], a *restricted Bayes classifier* modifies a Bayes

- A. Fujino and N. Ueda are with NTT Communication Science Laboratories, NTT Corporation, 2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan. E-mail: {a.fujino, ueda}@cslab.kecl.ntt.co.jp.
- K. Saito is with the School of Administration and Informatics, University of Shizuoka, 52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan. E-mail: k-saito@u-shizuoka-ken.ac.jp.

Manuscript received 5 June 2006; revised 28 Dec. 2006; accepted 25 Apr. 2007; published online 1 June 2007.
Recommended for acceptance by L. Kuncheva.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0418-0606.
Digital Object Identifier no. 10.1109/TPAMI.2007.70710.

optimal classifier based on maximum margin classification. It was shown that the hybrid classifier improved the generalization performance when the training set contained samples with missing feature values, but the missing label problem was not considered. In [12], for feature vectors structured by R components, the joint probability of each component is modeled individually, and these component models are combined with *weight* determined by maximizing the class posterior likelihood. That is, their hybrid classifier is constructed by a *discriminative* combination of *generative* models. They showed experimentally that the hybrid classifier performed better than or similar to pure generative and discriminative classifiers.

We propose a hybrid approach to semisupervised classifier design. In our formulation, a generative model is trained by using labeled samples. When the number of labeled samples is small, the class boundary provided by the trained generative model is often far from being the most appropriate one. That is, the trained generative model often has a high bias that results from the small number of labeled samples. To mitigate the effect of the bias associated with the trained generative model on the classification performance, we introduce a *bias correction model* that belongs to the same model family as the trained generative model. Then, by discriminatively combining these models based on the *maximum entropy* (ME) principle [13], we construct a classifier, called a *hybrid classifier*. The bias correction model is trained by using unlabeled samples.

We presented the basic idea of the hybrid approach in [14], [15], but we did not provide any explanation of the parameter estimation method. In this paper, we present a refined parameter estimation method for our hybrid classifier, where the bias correction model is trained to maximize the sum of the discriminative function values of unlabeled samples provided by the classifier. We expect this training to reduce the bias of the classifier that results from there being only a few labeled samples. The parameter of the bias correction model is estimated with the help of the EM algorithm. Then, we confirm experimentally the effect of bias correction by using unlabeled samples on the performance of our hybrid classifier.

The organization of the paper is organized as follows: In Section 2, we review conventional probabilistic generative and discriminative approaches to a semisupervised classifier design. In Section 3, we present our formulation for designing a semisupervised classifier based on a hybrid generative and discriminative approach. We also present a method for applying our hybrid approach to text classification problems by using naive Bayes (NB) models as the generative and bias correction models. In Section 4, using four test collections, we show experimentally that the generalization ability of our hybrid classifier is improved by using unlabeled samples for training and that our hybrid approach is particularly useful when generative and discriminative approaches exhibit comparable levels of performance. We also show the usefulness of the hybrid approach in terms of processing time. In Section 5, using an artificial data set, we show experimentally the effect of a massive number of unlabeled samples on the performance of our hybrid classifier. We also show the effect of using unlabeled samples whose distribution is different from that of labeled samples. Section 6 provides the conclusion.

2 Conventional Approaches

2.1 Semisupervised Learning

In multiclass (K classes) and single-labeled classification problems, one of the K classes $y \in \{1, \ldots, k, \ldots, K\}$ is assigned to a feature vector $\boldsymbol{x}$ by a classifier. In semisupervised learning settings, the classifier is trained on both labeled sample set $D_l = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$ and unlabeled sample set $D_u = \{\boldsymbol{x}_m\}_{m=1}^{M}$. Let $D = \{D_l, D_u\}$ represent a training sample set. Usually, M is much greater than N. We require a framework that will allow us to incorporate unlabeled samples without class labels y into classifiers. First, we briefly review the conventional probabilistic approaches.

2.2 Generative Approach

Generative classifiers learn a joint probability model $p(\boldsymbol{x}, y; \boldsymbol{\theta}_y)$, where $\Theta = \{\boldsymbol{\theta}_k\}_{k=1}^{K}$ is a set of model parameters over all classes. The class posteriors $P(y = k|\boldsymbol{x}; \Theta)$ for all classes are computed using Bayes rule after parameter estimation. The class label of $\boldsymbol{x}$ is determined as being y that maximizes $P(y = k|\boldsymbol{x}; \Theta)$. The joint probability model is designed according to classification tasks, for example, as a multinomial distribution model for text classification or as a Gaussian model for continuous feature vectors.

In the probabilistic framework, unlabeled samples are dealt with as the missing class labels in mixture models [16]. That is, $\boldsymbol{x}_m \in D_u$ is drawn from the marginal generative distribution $p(\boldsymbol{x}; \Theta) = \sum_{k=1}^{K} p(\boldsymbol{x}, k; \boldsymbol{\theta}_k)$. Model parameter Θ is computed by maximizing the posterior $p(\Theta|D)$ (maximum a posteriori (MAP) estimation). According to Bayes rule $p(\Theta|D) \propto p(D|\Theta)p(\Theta)$, the objective function of MAP estimation is given by

$$\begin{aligned} J_1(\Theta) = & \sum_{n=1}^{N} \log p(\boldsymbol{x}_n, y_n; \boldsymbol{\theta}_{y_n}) \\ & + \sum_{m=1}^{M} \log \sum_{k=1}^{K} p(\boldsymbol{x}_m, k; \boldsymbol{\theta}_k) + \log p(\Theta). \end{aligned} \tag{1}$$

Here, $p(\Theta)$ is a prior probability distribution of Θ. The value of Θ that maximizes $J_1(\Theta)$ is obtained by using the Expectation-Maximization (EM) algorithm [16].

The estimation of Θ is affected by the number of unlabeled samples used with the labeled samples. In other words, when $N \ll M$, model parameter Θ is estimated as almost unsupervised clustering because the second term on the right-hand side of (1) becomes much more dominant than the first term. This indicates that training the model by using unlabeled samples might not be useful in terms of classification accuracy if the mixture model assumptions are not true for actual classification tasks. As previously reported [17], using unlabeled samples can degrade the classification performance when there is a large difference between actual and assumed models. To mitigate the problem, a weighting parameter λ was introduced (EM-λ), which reduces the contribution of the unlabeled samples to the parameter estimation [1]. The weighting parameter $\lambda \in [0, 1]$ is multiplied in the second term on the right-hand side of (1). The parameter value is determined by a cross validation so that as far as possible, the leave-one-out labeled samples are correctly classified.

2.3 Discriminative Approach

Discriminative classifiers directly model class posterior probabilities $P(y|x)$ for all classes. In multinomial logistic regression (MLR) [18], the class posterior probabilities are modeled as

$$P(y=k|\boldsymbol{x};W) = \frac{\exp(\boldsymbol{w}_k \cdot \boldsymbol{x})}{\sum_{k'=1}^{K} \exp(\boldsymbol{w}_{k'} \cdot \boldsymbol{x})}, \tag{2}$$

where $W = \{\boldsymbol{w}_k\}_{k=1}^K$ is a set of unknown model parameters. $\boldsymbol{w}_k \cdot \boldsymbol{x}$ represent the inner product of $\boldsymbol{w}_k$ and $\boldsymbol{x}$.

MER was introduced as one way of incorporating unlabeled samples in discriminative classifiers [2]. This method is based on the empirical knowledge that classes should be well separated to take advantage of the unlabeled samples because the asymptotic information content of unlabeled samples decreases as classes overlap. Conditional entropy is used as a measure of class overlap. By minimizing the conditional entropy, the classifier is trained to separate unlabeled samples as well as possible.

Applying MER to MLR, we estimate W to maximize the following conditional log likelihood and regularizer

$$\begin{aligned} J_2(W) = &\sum_{n=1}^{N} \log P(y_n|\boldsymbol{x}_n;W) \\ &+ \lambda \sum_{m=1}^{M} \sum_{k=1}^{K} P(k|\boldsymbol{x}_m;W) \log P(k|\boldsymbol{x}_m;W) \\ &+ \log p(W). \end{aligned} \tag{3}$$

Here, λ is a weighting parameter and $p(W)$ is a prior probability distribution of W.

3 Hybrid Approach

As mentioned in Section 1, we propose a hybrid classifier based on the discriminative combination of generative and bias correction models. In this section, we present our formulation of the hybrid classifier and our parameter estimation method.

3.1 Generative Model and Bias Correction Model

We first design a class conditional generative model $p(\boldsymbol{x}|k;\boldsymbol{\theta}_k)$ for data samples $\boldsymbol{x}$ that belong to the kth class, where $\Theta = \{\boldsymbol{\theta}_k\}_{k=1}^K$ denotes a set of model parameters over all classes. In our formulation, the generative model is trained by using a set of labeled samples D_l. Θ is computed using the MAP estimation: $\hat{\Theta} = \arg\max_\Theta\{\log p(D_l|\Theta) + \log p(\Theta)\}$. Assuming that Θ is independent of class probability $P(y=k)$, we can derive the objective function for Θ estimation as

$$J(\Theta) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n|y_n;\boldsymbol{\theta}_{y_n}) + \log p(\Theta). \tag{4}$$

Here, $p(\Theta)$ is a prior probability distribution of Θ.

In semisupervised learning settings, the number of labeled samples is often small. Then, the classifier obtained by using the trained generative model often provides a class boundary that is far from being the most appropriate. That is, the trained generative model often has a high bias. To obtain a classifier with a smaller bias, we newly introduce another class conditional generative model, called the *bias correction model*. The bias correction model belongs to the same model family as the generative model, but the parameter set $\Psi = \{\boldsymbol{\psi}_k\}_{k=1}^K$ of the bias correction model is different from Θ. We construct our hybrid classifier by combining the generative and bias correction models to mitigate the effect of the bias associated with the trained generative model.

3.2 Discriminative Combination

We define our hybrid classifier by using the class posterior probability distribution derived from a discriminative combination of the generative and bias correction models. The combination is provided on the basis of the ME principle [13].

The ME principle prefers the most uniform probability distributions 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 $\hat{p}(\boldsymbol{x},k) = \sum_{n=1}^{N} I_{\boldsymbol{x}_n}(\boldsymbol{x}) I_{y_n}(k)/N$ of labeled samples as

$$\begin{aligned} &\sum_{\boldsymbol{x},k} \hat{p}(\boldsymbol{x},k) \log p(\boldsymbol{x}|k;\hat{\boldsymbol{\theta}}_k) \\ &= \sum_{\boldsymbol{x},k} \hat{p}(\boldsymbol{x}) R(k|\boldsymbol{x}) \log p(\boldsymbol{x}|k;\hat{\boldsymbol{\theta}}_k), \end{aligned} \tag{5}$$

where $\hat{p}(\boldsymbol{x}) = \sum_{n=1}^{N} I_{\boldsymbol{x}_n}(\boldsymbol{x})/N$ is the empirical distribution of $\boldsymbol{x}$. Here, $I_{\boldsymbol{x}_n}(\boldsymbol{x})$ is an indicator function, where $I_{\boldsymbol{x}_n}(\boldsymbol{x}) = 1$ if $\boldsymbol{x} = \boldsymbol{x}_n$; otherwise, $I_{\boldsymbol{x}_n}(\boldsymbol{x}) = 0$. The equation of the constraint for $\log p(\boldsymbol{x}|k;\boldsymbol{\psi}_k)$ can be represented in the same form as (5). We also restrict $R(k|\boldsymbol{x})$ so that it has the same class probability, as seen in the labeled samples, such that

$$\sum_{\boldsymbol{x}} \hat{p}(\boldsymbol{x},k) = \sum_{\boldsymbol{x}} \hat{p}(\boldsymbol{x}) R(k|\boldsymbol{x}), \forall k. \tag{6}$$

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

$$\begin{aligned} &R(k|\boldsymbol{x};\hat{\Theta},\Psi,\Gamma) \\ &= \frac{p(\boldsymbol{x}|k;\hat{\boldsymbol{\theta}}_k)^{\gamma_1} p(\boldsymbol{x}|k;\boldsymbol{\psi}_k)^{\gamma_2} e^{\mu_k}}{\sum_{k'=1}^{K} p(\boldsymbol{x}|k';\hat{\boldsymbol{\theta}}_{k'})^{\gamma_1} p(\boldsymbol{x}|k';\boldsymbol{\psi}_{k'})^{\gamma_2} e^{\mu_{k'}}}, \end{aligned} \tag{7}$$

where $\Gamma = (\gamma_1, \gamma_2, \{\mu_k\}_{k=1}^K)$ is a set of Lagrange multipliers. γ_1 and γ_2 provide combination weights for the generative and bias correction models, and μ_k provides a bias for the kth class.

The distribution $R(k|\boldsymbol{x};\hat{\Theta},\Psi,\Gamma)$ gives us the formulation of a discriminative classifier that consists of the generative and bias correction models. In a special case, $R(k|\boldsymbol{x};\hat{\Theta},\Psi,\Gamma)$ is reduced to the class posterior probability distribution derived from the generative model $p(\boldsymbol{x}|k;\hat{\boldsymbol{\theta}}_k)$ by using Bayes rule. Actually, if $\gamma_1 = 1$, $\gamma_2 = 0$, and $P(k) = e^{\mu_k}, \forall k$, then $R(k|\boldsymbol{x};\hat{\Theta},\Psi,\Gamma)$ is consistent with $P(k|\boldsymbol{x};\Theta) \propto p(\boldsymbol{x}|k;\hat{\boldsymbol{\theta}}_k)P(k)$. We employ $R(k|\boldsymbol{x};\hat{\Theta},\Psi,\Gamma)$ as the class posterior probability distribution of our hybrid classifier.

According to the ME principle, the solution of Γ in (7) is the same as the Γ that maximizes the log likelihood for $R(k|\boldsymbol{x};\hat{\Theta},\Psi,\Gamma)$ of labeled samples $(\boldsymbol{x}_n, y_n) \in D_l$ [13], [19]. However, D_l is also used to estimate Θ. Using the same labeled samples for both Γ and Θ may lead to the bias estimation of Γ. Thus, a leave-one-out cross validation of the labeled samples is used for the estimation of Γ, as applied in [12]. Let $\hat{\Theta}^{(-n)}$ be the generative model parameter estimated by using all the labeled samples except $(\boldsymbol{x}_n, y_n)$. The objective function of Γ then becomes

$$F(\Gamma|\Psi) = \sum_{n=1}^{N} \log R\Big(y_n|\boldsymbol{x}_n; \hat{\Theta}^{(-n)}, \Psi, \Gamma\Big) + \log p(\Gamma). \tag{8}$$

Here, $p(\Gamma)$ is a prior probability distribution of Γ. We used the Gaussian prior [20] as

$$p(\Gamma) \propto \prod_{j=1}^{2} \exp\left\{-\frac{(\gamma_j - a_j)^2}{2\sigma_j^2}\right\} \prod_{k=1}^{K} \exp\left(-\frac{\mu_k^2}{2\rho_k^2}\right), \tag{9}$$

where a_j, σ_j, and ρ_k are hyperparameters in the Gaussian prior. We can compute an estimate of Γ to maximize $F(\Gamma|\Psi)$ under a fixed Ψ by using the limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm [21], which is a quasi-Newton method. In this computation, global convergence is guaranteed, since $F(\Gamma|\Psi)$ is a concave function of Γ. In practice, we compute Γ to maximize $F(\Gamma|\Psi)$ under the constraint of $\gamma_2 \geq 0$ because $\gamma_2 < 0$ indicates the negative use of information supplied by the bias correction model.

3.3 Learning of Bias Correction Model Parameter

In our formulation, the parameter Ψ of the bias correction model is trained with unlabeled samples to reduce the bias that results from a small number of labeled samples. According to $R(k|\boldsymbol{x}; \hat{\Theta}, \Psi, \Gamma)$, as shown in (7), the class label y of a feature vector $\boldsymbol{x}$ is determined as the k that maximizes the discriminative function

$$g_k(\boldsymbol{x}; \boldsymbol{\psi}_k) = p(\boldsymbol{x}|k; \hat{\boldsymbol{\theta}}_k)^{\gamma_1} p(\boldsymbol{x}|k; \boldsymbol{\psi}_k)^{\gamma_2} e^{\mu_k}. \tag{10}$$

Here, when the $g_k(\boldsymbol{x}; \boldsymbol{\psi}_k)$ values for all classes are small, the classification result for $\boldsymbol{x}$ would not be reliable because $g_k(\boldsymbol{x}; \boldsymbol{\psi}_k)$ is almost the same for all classes. Thus, we expect our classifier to provide a large $g_k(\boldsymbol{x}; \boldsymbol{\psi}_k)$ difference between classes for unseen samples by estimating Ψ that maximizes the sum of the discriminative functions of unlabeled samples

$$G(\Psi|\Gamma) = \sum_{m=1}^{M} \log \sum_{k=1}^{K} g_k(\boldsymbol{x}_m; \boldsymbol{\psi}_k) + \log p(\Psi), \tag{11}$$

where $p(\Psi)$ is a prior probability distribution of Ψ.

If Γ is known, then we can estimate Ψ that provides the local maximum of $G(\Psi|\Gamma)$ around the initialized value of Ψ by using an iterative computation such as the EM algorithm. Let $\Psi^{(t)}$ be the estimated Ψ in the (t)th step. Then, using $\Psi^{(t)}$, we estimate Ψ in the $(t+1)$th step $\Psi^{(t+1)}$ to maximize the Q function

$$\begin{aligned} &Q\Big(\Psi^{(t+1)}, \Psi^{(t)}|\Gamma\Big) \\ =&\gamma_2 \sum_{m=1}^{M} \sum_{k=1}^{K} R\Big(k|\boldsymbol{x}_m; \hat{\Theta}, \Psi^{(t)}, \Gamma\Big) \log p\Big(\boldsymbol{x}_m|k; \boldsymbol{\psi}_k^{(t+1)}\Big) \\ &+ \log p\Big(\Psi^{(t+1)}\Big). \end{aligned} \tag{12}$$

We can obtain the estimate of Ψ by iteratively performing this update while $G(\Psi|\Gamma)$ is hill climbing (see Appendix A for the derivation of (12)).

```
Given training set: $D_l = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ and $D_u = \{\boldsymbol{x}_m\}_{m=1}^M$
1. Initialize $\Psi^{(0)}$, $t \leftarrow 0$.
2. Compute $\hat{\Theta}$ and $\hat{\Theta}^{(-n)}$, $\forall n$, using (4).
3. Compute $\Gamma^{(0)}$ using (8) under fixed $\Psi^{(0)}$ and $\hat{\Theta}^{(-n)}$.
4. Perform the following until $d(t) < \epsilon$ (see (13)).
   • Compute $\Psi^{(t+1)}$ using (12) under fixed $\Gamma^{(t)}$ and $\hat{\Theta}$.
   • Compute $\Gamma^{(t+1)}$ using (8) under fixed $\Psi^{(t+1)}$ and $\hat{\Theta}^{(-n)}$.
   • $t \leftarrow t + 1$.
5. Output a classifier $R(k|\boldsymbol{x}; \hat{\Theta}, \Psi^{(t)}, \Gamma^{(t)})$.
```

Fig. 1. Algorithm of learning model parameters in our hybrid approach.

However, Γ is also an unknown parameter that should be estimated using (8), with Θ and Ψ fixed. That is, the estimations of Ψ and Γ are mutually dependent. As a solution to our parameter estimation, we search for the Ψ and Γ that maximize $F(\Gamma|\Psi)$ and $G(\Psi|\Gamma)$ simultaneously. For this search, we compute Ψ and Γ by maximizing the objective functions shown in (12) and (8) iteratively and alternately. After estimating Θ by using (4), we first provide an initialized value $\Psi^{(0)}$ and estimate Γ by using (8) with $\Psi^{(0)}$. Next, we estimate $\Psi^{(1)}$ by using (12) with the estimate of Γ. Then, we estimate Γ with $\Psi^{(1)}$ again. We update Γ and Ψ iteratively until a certain convergence criterion, $d(t) < \epsilon$, is met,

$$d(t) = \sum_{k=1}^{K} \frac{||\boldsymbol{\psi}_k^{(t+1)} - \boldsymbol{\psi}_k^{(t)}||}{||\boldsymbol{\psi}_k^{(t)}||} + \frac{||\Gamma^{(t+1)} - \Gamma^{(t)}||}{||\Gamma^{(t)}||}. \tag{13}$$

Here, $\|\psi_k^{(t)}\|$ represents the level-2 (L2)-norm of $\boldsymbol{\psi}_k^{(t)}$. We summarize the algorithm for estimating these model parameters in Fig. 1.

Our learning algorithm shown in Fig. 1 estimates Ψ and Γ iteratively and alternately to maximize two objective functions $F(\Gamma|\Psi)$ and $G(\Psi|\Gamma)$ simultaneously. In each step of the iterative computation, Ψ and Γ are updated alternately to provide large objective function values. The value of Γ providing a global maximum of $F(\Gamma|\Psi)$ is computed easily under an arbitrary fixed value in the Ψ domain, since $F(\Gamma|\Psi)$ is a concave function of Γ. However, if $G(\Psi|\Gamma)$ is not a concave function of Ψ, then there is no guarantee that such an alternate gradient method as our learning algorithm can lead us to a local optimal point in Ψ and Γ from an initialized parameter value. When the iterative computation performed by our learning algorithm does not reach a local optimal point, the computation may cause the parameter estimates to oscillate, and thus the convergence criterion for our parameter estimation would not be satisfied. To deal with this convergence problem, we stopped the iterative computation when t reached its upper limit in our experiments described in Section 4. We observed such oscillation when training our hybrid classifier on some experimental settings but confirmed that our hybrid performed better than other semisupervised classifiers.

3.4 Application of Hybrid Approach to Text Classification

For text classification problems, we employed NB models as the generative model $p(\boldsymbol{x}|k; \boldsymbol{\theta}_k)$ and bias correction model $p(\boldsymbol{x}|k; \boldsymbol{\psi}_k)$ by using the independent word-based representation known as the Bag-of-Words (BOW) representation. Let $\boldsymbol{x} = (x_1, \ldots, x_i, \ldots, x_V)$ represent the feature vector of a document, where x_i denotes the frequency of the ith word in the document, and V denotes the number of vocabulary words included in a text data set. In an NB model, the probability distribution of document $\boldsymbol{x}$ in the $k\mathrm{th}$ class is regarded as a multinomial distribution

$$p(\boldsymbol{x}|k;\boldsymbol{\theta}_k) \propto \prod_{i=1}^{V} (\theta_{ki})^{x_i}. \tag{14}$$

Here, $\theta_{ki} > 0$, and $\sum_{i=1}^{V} \theta_{ki} = 1$. θ_{ki} is the probability that the $i\mathrm{th}$ word appears in a document belonging to the $k\mathrm{th}$ class. $p(\boldsymbol{x}|k;\boldsymbol{\psi}_k)$ is also given the same distribution form as $p(\boldsymbol{x}|k;\boldsymbol{\theta}_k)$. When applying NB models to our hybrid classifier, we used feature vectors normalized with vector size $|\boldsymbol{x}| = \sum_{i=1}^{V} x_i$.

For the MAP estimation of $\boldsymbol{\theta}_k$, as the prior $p(\boldsymbol{\theta}_k)$ in (4), we use a Dirichlet prior $p(\boldsymbol{\theta}_k) \propto \prod_{i=1}^{V} (\theta_{ki})^{\xi_k - 1}$, where $\xi_k (> 1)$ represents a hyperparameter. A Dirichlet prior is also used for $p(\Psi)$ in (11). Let $\{\boldsymbol{x}_{n_k}\}_{n_k=1}^{N_k}$ represent the normalized feature vectors of labeled samples that belong to the $k\mathrm{th}$ class. Then, the estimate of θ_{ki} is computed as

$$\hat{\theta}_{ki} = \frac{\sum_{n_k=1}^{N_k} x_{n_k i} + \xi_k - 1}{N_k + V(\xi_k - 1)}. \tag{15}$$

The estimate of ψ_{ki} in the $(t+1)\mathrm{th}$ step is computed as

$$\psi_{ki}^{(t+1)} = \frac{\gamma_2 \sum_{m=1}^{M} R(k|\boldsymbol{x}_m; \hat{\Theta}, \Psi^{(t)}, \Gamma^{(t)}) x_{mi} + \eta_k - 1}{\gamma_2 \sum_{m=1}^{M} R(k|\boldsymbol{x}_m; \hat{\Theta}, \Psi^{(t)}, \Gamma^{(t)}) + V(\eta_k - 1)}, \tag{16}$$

where η_k is a hyperparameter of Dirichlet prior $p(\Psi)$.

To estimate $\boldsymbol{\theta}_k$, we tune ξ_k to maximize the sum of the log likelihood computed with a leave-one-out cross validation of the labeled samples [22]

$$\begin{aligned} L(\xi_k) &= \sum_{n_k=1}^{N_k} \log P\left(\boldsymbol{x}_{n_k}|k; \hat{\boldsymbol{\theta}}_k^{(-n_k)}\right) \\ &= \sum_{n_k=1}^{N_k} \sum_{i=1}^{V} x_{n_k i} \log \hat{\theta}_{ki}^{(-n_k)} \end{aligned} \tag{17}$$

because we confirmed that this tuning was practically useful for classification. Here, $\hat{\boldsymbol{\theta}}_k^{(-n_k)}$ $(\hat{\theta}_{ki}^{(-n_k)})$ is the estimate of $\boldsymbol{\theta}_k$ (θ_{ki}) computed by training samples except $\boldsymbol{x}_{n_k}$. This tuning is executed with the help of the EM algorithm [16] (see Appendix B for details).

4 Experimental Evaluation Using Real Text Data

4.1 Test Collections

To evaluate our proposed hybrid classifier empirically, we used four test collections that have often been employed as benchmark tests for classifiers in text classification tasks. The first collection is the Reuters-21578 data set (Reuters), which consists of 135 topic categories from the Reuters newswire [23]. The 10 most frequently occurring categories have often been used for benchmark tests, and thus, we constructed a subset by selecting articles belonging to one of the 10 categories. For single-labeled classification tasks, we removed multilabeled articles. Since two of the 10 categories contained few articles, we used eight categories—*acq*, *crude*, *earn*, *grain*, *interest*, *money-fx*, *ship*, and *trade*—that contained many single-labeled articles. Reuters divides the articles into two groups by point in time, and there were 6,182 earlier articles and 2,430 later articles in the subset. In our experiments, the later articles were used for test samples, and the earlier articles were selected as labeled or unlabeled samples. We removed vocabulary words included either in the stoplist [24] or in only one article. We used 11,822 vocabulary words to represent feature vectors of articles in the data set.

The second collection is the WebKB data set (WebKB), which contains Web pages from universities. This data set consists of seven categories, and each page belongs to one of the categories. Following the setup in [19], we used only four categories: *course*, *faculty*, *project*, and *student*. There were 4,199 Web pages in these categories. We removed tags and links from the pages, as well as vocabulary words in the same way as for Reuters. We used 18,525 vocabulary words in the data set.

The third collection is the 20 newsgroups data set (20news), which consists of 20 different UseNet discussion groups. Following the setup in [19], we used only five groups: *comp.**. There were 4,881 articles in the groups. We removed vocabulary words in the same way as for Reuters. We used 19,383 vocabulary words in the data set.

The fourth collection is the Cora data set (Cora),[1] which contains more than 30,000 summaries of technical papers, and each paper belongs to one of the 70 groups. For our evaluation, we used 4,240 papers included in seven groups: */Artificial_Intelligence/Machine_Learning/**. Each paper contains an abstract and a citation list. We removed vocabulary words in the same way as for Reuters and removed works cited by only one paper. We used 9,190 vocabulary words and 13,282 cited works in the data set.

To apply our hybrid approach to the Cora data samples that consist of text and citation feature vectors $\boldsymbol{x}^t$ and $\boldsymbol{x}^c$ we assumed the text and citations to be independent of each other. Under this assumption, we designed the generative model of $\boldsymbol{x} = (\boldsymbol{x}^t, \boldsymbol{x}^c)$ in the $k\mathrm{th}$ class such as $p(\boldsymbol{x}|k;\boldsymbol{\theta}_k) = p(\boldsymbol{x}^t|k;\boldsymbol{\theta}_k^t)p(\boldsymbol{x}^c|k;\boldsymbol{\theta}_k^c)$. Here, $\boldsymbol{\theta}_k^t$ and $\boldsymbol{\theta}_k^c$ represent the text and citation model parameters, respectively. We employed NB models individually as$p(\boldsymbol{x}^t|k;\boldsymbol{\theta}_k^t)$ and $p(\boldsymbol{x}^c|k;\boldsymbol{\theta}_k^c)$.

4.2 Effect of Using Unlabeled Samples for Training

We examined the effect of using unlabeled samples for training our hybrid classifier on its generalization ability. As the evaluation measure for examining the performance of trained classifiers, we employed classification accuracy, which refers to how many test samples were correctly classified. The classification accuracy AC is evaluated as $AC = T/S$, where T is the number of correctly classified test samples, and S is the total number of test samples.

In multiclass and single-labeled classification problems, the classification accuracy is equivalent to the microaveraged F-measure [25], which has been used to evaluate classifiers in binary or multilabeled classification problems (cf. [26]). The microaveraged F-measure FM is evaluated as $FM = 2/(1/PR + 1/RE)$, where PR and RE represent the microaveraged precision and recall such as $PR = \sum_{k=1}^{K} T_k / \sum_{k=1}^{K} p_k$ and $RE = \sum_{k=1}^{K} T_k / \sum_{k=1}^{K} r_k$. Here, T_k represents the number of test samples whose true class labels are k and that were predicted correctly by the classifier. p_k represents the number of test samples whose class labels are predicted as k by the classifier. r_k represents the number of test samples whose true class labels are k. Since $\sum_{k=1}^{K} p_k = \sum_{k=1}^{K} r_k = S$ and $\sum_{k=1}^{K} T_k = T$

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

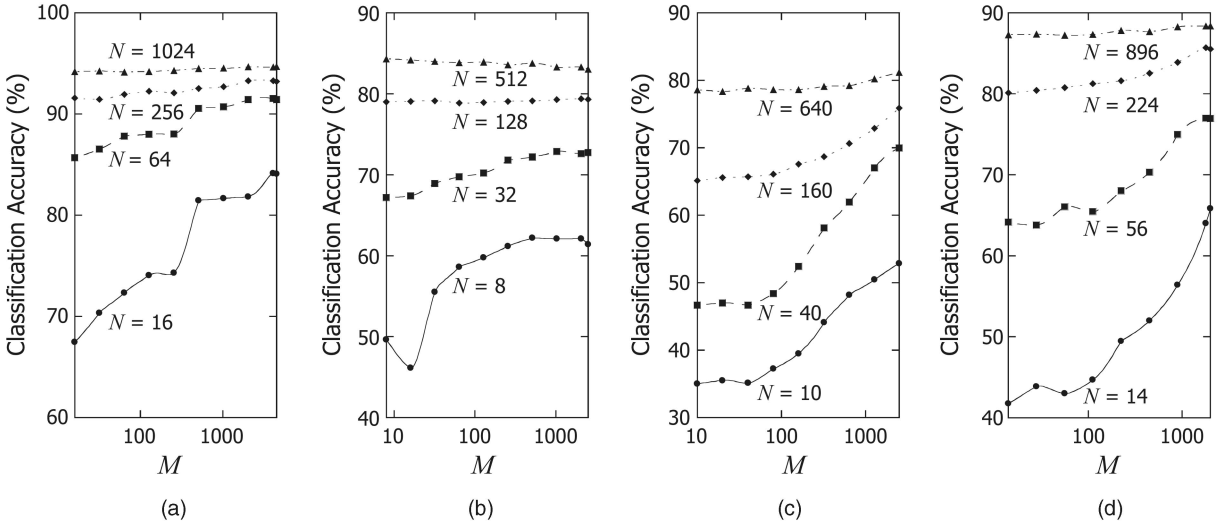


Fig. 2. Classification accuracies (in percent) with our proposed hybrid classifier trained on M unlabeled samples with N fixed labeled samples. (a) Reuters. (b) WebKB. (c) 20news. (d) Cora.

in multiclass and single-labeled classification problems, FM is reduced to $FM = T/S$.

Fig. 2 shows the average classification accuracies of 10 experiments undertaken with our hybrid classifier trained by using M unlabeled samples with N fixed labeled samples. In each experiment, the classification accuracies were examined using test samples that were different from the labeled and unlabeled samples. The labeled, unlabeled, and test samples were randomly selected from each data set. With Reuters, 2,430 later articles were used as test samples. With WebKB, 20news, and Cora, 1,000 samples were selected as test samples. After selecting the test samples, we selected labeled and unlabeled samples from the remaining samples in each data set.

As shown in Fig. 2, the classification accuracies for a large number of unlabeled samples tend to be higher than those for a small number of unlabeled samples with a fixed number of labeled samples, except $N = 512$ on WebKB. Using a large number of unlabeled samples for training our hybrid classifier greatly improved the classification performance when the number of labeled samples was small. We confirmed that unlabeled samples were beneficial for training with our hybrid approach, especially when the number of labeled samples was insufficiently large to obtain good classification performance.

4.3 Comparison with Generative and Discriminative Approaches

4.3.1 Experimental Settings

We compared our hybrid classifier with conventional semisupervised generative and discriminative classifiers: an NB classifier with EM-λ (NB/EM-λ) [1] and an MLR classifier with an MER (MLR/MER) [2]. We also compared the hybrid classifier with two *supervised* classifiers: NB and MLR classifiers [19]. NB and MLR were trained only on labeled samples.

In our experiments, for NB/EM-λ, the value of weighting parameter λ was set by maximizing the leave-one-out cross-validation classification accuracy of the labeled samples, following the method described in [1]. Note that in our experiments, we selected the value from 14 candidate values of {0.01, 0.05, 0.1, 0.2, 0.25, 0.3, 0.4, 0.5, 0.6, 0.7, 0.75, 0.8, 0.9, 1.0} to save computation time. However, these candidate values were carefully selected via preliminary experiments. We used a Dirichlet distribution for $p(\Theta)$, and its hyperparameter was set in a similar manner to λ.

For MLR/MER, the value of weighting parameter λ in (3) was selected from 16 candidate values of $\{\{0.1 \times 10^{-n}, 0.2 \times 10^{-n}, 0.5 \times 10^{-n}\}_{n=0}^{4}, 1\}$ that were carefully selected via preliminary experiments. For a fair comparison of the methods, the value of λ in MLR/MER should also be determined using training samples, for example, using a cross validation of labeled samples [2]. We determined the λ value that gave the best classification performance for *test* samples to examine the *potential ability* of MLR/MER because the computation cost of tuning λ was very high. For both MLR and MLR/MER, we fixed values for the hyperparameters in the Gaussian prior that gave a high average classification accuracy for the test samples to observe the potential ability of the method.

To compare these classifiers, we employed the average classification accuracies of 10 experiments. In each experiment, 2,430 later articles from Reuters were used as the test samples, and 1,000 test samples were selected randomly from WebKB, 20news, and Cora. Four thousand five hundred, 2,500, 2,500, and 2,000 unlabeled samples were selected randomly from Reuters, WebKB, 20news, and Cora, respectively. Various numbers of labeled samples were selected randomly from the other samples for each data set. With the random selection, we obtained 10 different sample sets per sample number. For each classifier, we calculated the average of 10 classification accuracies, which we examined by using the different sample sets.

4.3.2 Classification Performance

Table 1 shows the average classification accuracies of 10 experiments with the five classifiers in Reuters, WebKB, 20news, and Cora. Each number in parentheses in the table denotes the standard deviation of the 10 experiments. N and M represent the number of labeled and unlabeled samples. Asterisks show that the difference between the average classification accuracies of our hybrid classifier and the

TABLE 1
Classification Accuracies (in Percent) with Five Classifiers Trained on N Labeled Samples with M Fixed Unlabeled Samples

		Semi-supervised ($M = 4500$)			Supervised		Full-labeled	
N	N/M	Hybrid	NB/EM-λ	MLR/MER	NB	MLR	NB	MLR
16	.0036	84.1 (3.7)	**86.9** (2.6)	74.1 (8.3)*	71.1 (4.4)*	67.6 (7.9)*	94.8 (0.1)	95.3 (0.1)
32	.0071	**89.4** (1.6)	89.0 (2.9)	82.5 (3.9)*	80.4 (1.8)*	81.0 (3.5)*	94.8 (0.1)	95.3 (0.1)
64	.014	**91.4** (0.8)	90.6 (3.1)	84.1 (4.6)*	85.1 (2.2)*	83.3 (4.6)*	94.9 (0.1)	95.3 (0.1)
128	.028	**92.8** (0.7)	92.0 (0.8)	89.3 (1.5)*	88.7 (0.6)*	88.2 (1.2)*	95.0 (0.1)	95.2 (0.1)
256	.057	**93.2** (0.9)	93.0 (1.0)	91.7 (0.8)*	90.3 (1.0)*	90.7 (0.7)*	95.0 (0.2)	95.3 (0.1)
512	.11	**94.1** (0.4)	93.4 (0.5)*	93.7 (0.3)*	92.9 (0.6)*	93.0 (0.4)*	95.0 (0.1)	95.4 (0.1)
1024	.23	94.7 (0.2)	94.3 (0.3)*	**94.9** (0.3)	94.1 (0.4)*	94.5 (0.4)	95.2 (0.1)	95.6 (0.2)

(a)

		Semi-supervised ($M = 2500$)			Supervised		Full-labeled	
N	N/M	Hybrid	NB/EM-λ	MLR/MER	NB	MLR	NB	MLR
8	.0032	61.4 (5.9)	**63.4** (8.5)	54.0 (4.0)*	53.0 (7.9)*	53.6 (4.3)*	83.8 (0.9)	92.1 (0.5)
16	.0064	66.8 (5.0)	**69.4** (2.6)	54.1 (5.6)*	60.2 (3.7)*	54.0 (5.1)*	83.8 (1.0)	92.0 (0.5)
32	.013	**72.7** (3.3)	**72.7** (2.5)	63.6 (4.5)*	69.2 (2.8)*	63.5 (4.3)*	83.8 (1.0)	92.1 (0.6)
64	.026	**76.2** (2.1)	75.4 (2.4)	72.4 (2.5)*	73.3 (0.9)*	72.1 (2.2)*	83.7 (0.9)	92.0 (0.5)
128	.051	**79.3** (1.5)	76.9 (2.2)*	78.4 (2.1)	76.9 (2.1)*	78.3 (2.1)	83.8 (1.1)	92.2 (0.4)
256	.10	81.3 (1.5)	79.7 (1.9)*	**84.1** (1.6)*	79.9 (1.4)*	83.6 (1.6)*	83.9 (1.1)	92.2 (0.5)
512	.20	83.1 (1.7)	82.2 (1.4)*	**88.5** (1.3)*	82.5 (1.3)	87.4 (1.3)*	84.2 (1.1)	92.3 (0.6)

(b)

		Semi-supervised ($M = 2500$)			Supervised		Full-labeled	
N	N/M	Hybrid	NB/EM-λ	MLR/MER	NB	MLR	NB	MLR
10	.0040	**52.9** (13.2)	39.5 (9.0)*	41.0 (7.7)*	32.4 (6.1)*	37.5 (5.4)*	83.0 (1.6)	85.5 (0.9)
20	.0080	**64.3** (6.3)	50.3 (7.8)*	45.3 (4.9)*	41.3 (4.6)*	44.5 (4.3)*	82.9 (1.5)	85.5 (0.9)
40	.016	**70.0** (2.4)	56.1 (6.4)*	52.5 (4.9)*	47.4 (3.9)*	50.9 (3.6)*	83.1 (1.6)	85.5 (0.9)
80	.032	**73.1** (2.1)	61.3 (5.3)*	59.9 (2.9)*	53.4 (3.9)*	58.9 (2.3)*	83.1 (1.3)	85.7 (0.9)
160	.064	**75.9** (1.5)	67.2 (4.2)*	68.2 (2.5)*	61.1 (2.6)*	66.4 (1.9)*	83.0 (1.7)	85.9 (0.8)
320	.13	**78.4** (0.9)	70.8 (2.7)*	74.6 (1.4)*	68.8 (1.4)*	72.7 (1.1)*	83.5 (1.7)	86.0 (0.8)
640	.26	**81.2** (1.2)	75.8 (1.8)*	79.4 (1.1)*	74.9 (2.1)*	77.8 (1.1)*	84.2 (1.6)	86.5 (1.0)
1280	.51	**83.8** (1.1)	78.9 (2.1)*	82.5 (1.3)*	77.8 (1.6)*	81.2 (1.4)*	84.9 (1.5)	87.0 (1.2)

(c)

		Semi-supervised ($M = 2000$)			Supervised		Full-labeled	
N	N/M	Hybrid	NB/EM-λ	MLR/MER	NB	MLR	NB	MLR
14	.0070	**65.8** (7.8)	46.4 (11.0)*	45.2 (4.9)*	34.5 (7.5)*	41.2 (4.0)*	87.2 (1.0)	88.2 (0.9)
28	.014	**72.6** (6.5)	63.0 (5.7)*	54.2 (4.1)*	45.7 (4.3)*	50.9 (4.0)*	87.3 (0.9)	88.3 (1.0)
56	.028	**77.0** (4.0)	71.4 (4.6)*	63.4 (5.7)*	56.8 (3.0)*	61.5 (4.3)*	87.5 (1.1)	88.3 (0.9)
112	.056	**82.9** (2.1)	76.9 (1.6)*	69.9 (3.7)*	65.1 (1.5)*	69.1 (3.4)*	87.9 (0.9)	88.2 (1.0)
224	.11	**85.5** (0.8)	80.9 (1.4)*	78.2 (1.3)*	74.7 (1.7)*	76.7 (1.2)*	87.8 (1.3)	88.4 (1.1)
448	.22	**87.0** (1.3)	83.4 (1.1)*	83.0 (1.5)*	79.8 (1.1)*	81.6 (1.4)*	88.2 (1.2)	88.7 (1.0)
896	.45	**88.4** (1.0)	86.2 (1.2)*	86.3 (1.4)*	84.4 (0.9)*	85.5 (1.3)*	88.8 (0.8)	89.1 (1.0)

(d)

(a) Reuters. (b) WebKB. (c) 20news. (d) Cora.

classifiers used for comparison is significant ($p < 0.05$) in the Wilcoxon test, which has been used for statistical comparisons of the abilities of classifiers (cf. [27]). We also examined the classification accuracies of the NB and MLR classifiers trained by using all the $N + M$ samples as labeled samples and confirmed the potentiality of unlabeled samples for improving classification performance. These results are shown in the column headed "Full labeled" in Table 1.

In the supervised case, as reported in [10], generative (discriminative) classifiers provided better classification performance than the discriminative (generative) classifiers when the number of labeled samples was small (large). In our experiments using NB and MLR in supervised settings on Reuters and WebKB, we obtained similar results as those in [10].

However, in 20news and Cora, MLR outperformed NB, even when the number of training samples was small, which seems to be inconsistent with [10]. To investigate the result further, we analyze the fit of NB models to real data samples for each data set, since it was reported that the difference between the assumed generative models and the distribution of real data affected classification performance [17]. We examined the *normalized test perplexity (NTP)* of the trained NB model for each test collection. NTP is a measure of how well the estimated model fits the test samples $\{(\boldsymbol{x}_s, y_s)\}_{s=1}^{S}$ not used in the training and is defined by

$$NTP = \frac{1}{V}\exp\left(-\frac{\sum_{k=1}^{K}\sum_{s=1}^{S} I_{y_s}(k)\sum_{i=1}^{V} x_{si}\log\hat{\theta}_{ki}}{\sum_{s=1}^{S}\sum_{i=1}^{V} x_{si}} \right), \quad (18)$$

where $\hat{\theta}_{ki}$ is a parameter estimated using the training data, and $I_{y_s}(k)$ is a class indicator ($I_{y_s}(k) = 1$ if $k = y_s$; otherwise, $I_{y_s}(k) = 0$). A smaller NTP value means a better fit with the test samples. $NTP = 1$ for the random model, where

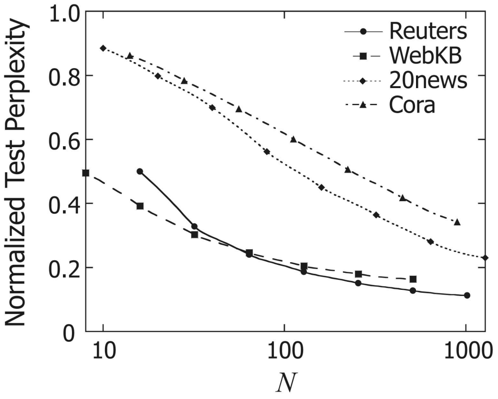


Fig. 3. NTPs of NB models trained on N labeled samples for four test collections.

$\hat{\theta}_{ki} = 1/V, \forall k, \forall i$. This means that the trained models, where $NTP < 1$ ($NTP > 1$), fit the test samples better (worse) than the random model. For NTP in Cora, we examined the NTPs of the NB models for text and citations and averaged their perplexities with weights based on the dimensions of the feature vectors for text and citations. As shown in Fig. 3, the NTP values in 20news and Cora were significantly larger than in Reuters and WebKB when the number of training samples was less than 1,000. This indicates that the NB generative model did not fit the test samples well when the training data size was small on 20news and Cora. In other words, if a smaller NTP had been obtained for a small N, NB would have outperformed MLR. Thus, in a supervised setting, generative classifiers can outperform discriminative classifiers when the test perplexities of the estimated generative models are good enough in small N settings. This finding should be included in conventional discussions on generative and discriminative classifiers in supervised settings.

We examined NB/EM-λ, MLR/MER, and our hybrid approach in semisupervised cases. First, the classification performance of NB/EM-λ tended to be better (worse) than that of MLR/MER for all data sets when N was small (large). That is, we confirmed that the characteristics of the generative and discriminative approaches in supervised learning also hold for semisupervised learning, which seems reasonable.

Second, we found that our hybrid approach outperformed NB/EM-λ when MLR outperformed NB in supervised cases. This result indicates that our hybrid approach takes advantage of the discriminative approach. However, the classification performance of our hybrid approach was slightly worse than that of NB/EM-λ when there were a small number of labeled samples for Reuters and WebKB. In these cases, the labeled sample sets used for training contained less than 1/10 of the vocabulary words of the data sets. Our hybrid classifier would be more overfitted to labeled samples existing in a small part of the feature space by the discriminative training than the NB/EM-λ classifier.

Finally, our hybrid approach outperformed MLR/MER, except when there were many labeled samples for Reuters and WebKB. This result is because the MLR/MER classifier tends to be overfitted to a small number of labeled samples. In contrast, our hybrid approach inherently has the characteristics of the generative model, whereby such an overfitting problem is mitigated. When many labeled samples are available such that the overfitting problem can be overcome, it would be natural for the discriminative approach to be better than our hybrid approach.

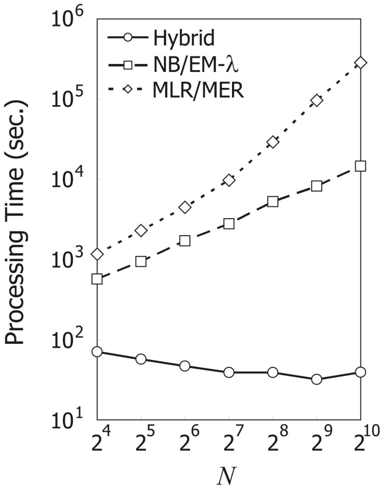


Fig. 4. Processing time for parameter estimation using N labeled samples and 4,500 fixed unlabeled samples on Reuters.

Our comparison of the three semisupervised approaches confirmed that our hybrid approach outperformed NB/EM-λ and MLR/MER when NB/EM-λ and MLR/MER performed similarly. Our hybrid approach was useful for obtaining better classifiers, especially when the classification performance of the generative and discriminative approaches was comparable.

4.3.3 Processing Time

We examined the processing time needed for training the three semisupervised classifiers, under the condition that the hyperparameters of all the classifiers were determined. Fig. 4 shows the average processing time for 10 experiments with Reuters. For a fair comparison of the classifiers, we estimated the processing time needed to determine λ for MLR/MER by using a leave-one-out cross validation of the labeled samples and plotted it in Fig. 4.

To train our hybrid classifier, we computed Ψ and Γ alternately and iteratively until $d(t) < 10^{-5}$ or $t = 200$. The size of Ψ is the product of the number of classes K and the dimension of the feature vectors, V. The size of Ψ is larger than that of Γ, which is $2 + K$. In our experiments, an average of 170 (66) iterative computations for estimating Ψ and Γ were required when $N = 16$ $(N = 1,024)$. The average number of iterative computations required for a small number of labeled samples was larger than that for a large number of labeled samples. As shown in Fig. 4, our hybrid classifier required a long training time when the number of labeled samples was small. This processing time would be the result of the iterative computations required for the parameter estimation.

In NB/EM-λ, Θ is estimated with the EM algorithm, under a fixed value of weight parameter λ. Θ is the same size as Ψ in our hybrid approach. In our experiments, Θ was computed iteratively until $\{J_1(\Theta^{(t+1)}) - J_1(\Theta^{(t)})\}/J_1(\Theta^{(t)}) < 10^{-4}$ or $t = 200$. An average of about 10 (4) iterative computations were required for estimating Θ when $N = 16$ $(N = 1,024)$. The average number of iterative computations required for a large number of labeled samples was smaller than that for a small number of labeled samples. These iterative computation results suggest that the computation cost of NB/EM-λ with a fixed λ is smaller than that of our hybrid approach. In our preliminary examination, we confirmed experimentally that estimating Θ for NB/EM-λ under a fixed λ required a shorter processing time than estimating Ψ and Γ for our hybrid classifier. However, λ is also a parameter that should be determined using training samples. To determine λ, we need to compute the estimates of Θ per λ candidate by using resampling techniques. When we employ a leave-one-out cross validation of labeled samples as the resampling technique, we need $C \times N$ estimations of Θ, where C represents the number of candidates for λ. Therefore, as shown in Fig. 4, the NB/EM-λ processing time tended to be proportional to the number of labeled samples N. This is why the repeated estimation of Θ to determine λ would lead to a longer processing time with NB/EM-λ than with our hybrid approach.

W for MLR/MEL is the same size as Θ for NB/EM-λ. W is also computed with a fixed value for weight parameter λ. To determine λ, we need to estimate W by using a cross validation of labeled samples with each λ candidate as with Θ for NB/EM-λ. In our experiments, MLR/MER required a longer processing time than NB/EM-λ. This suggests that learning with the MER may generally require a large number of training iterations due to its relatively high nonlinearity.

4.4 Effect of Bias Correction Model

In Section 4.3.2, we confirmed that our hybrid classifier outperformed the NB classifier with EM-λ. However, the number of generative model parameters used for our hybrid classifier was larger than that of the NB classifier, and this might have affected their classification performance. To confirm the effect of introducing a bias correction model, we also examined the classification performance of Mixture of Experts (MoE) classifiers [28]. In the MoE formulation, we design joint probability models $\{p(\boldsymbol{x}, k; \Theta)\}_{k=1}^{K}$ by using a mixture of multiple conditional probability models $\{p(\boldsymbol{x}|j; \boldsymbol{\theta}_j)\}_{j=1}^{J}$ belonging to the same model family such as

$$p(\boldsymbol{x}, k; \Theta) = \sum_{j=1}^{J} P(k|j) p(\boldsymbol{x}|j; \boldsymbol{\theta}_j) P(j), \forall k. \tag{19}$$

In our experiments, we examined MoE classifiers where $J = 2K$, since our hybrid classifier was designed by the combination of the $2K$ probability models that were generative and bias correction models $\{p(\boldsymbol{x}|k; \boldsymbol{\theta}_k), p(\boldsymbol{x}|k; \boldsymbol{\psi}_k)\}_{k=1}^{K}$. We compared our hybrid approach with MoE by using the same number of generative model parameters.

For MoE classifiers, we employed two models described in [28]: The partitioned mixture (PM) and the generalized mixture (GM) models. We call MoE classifiers with GM (PM) MoE-GM (MoE-PM). The GM model is represented as shown in (19). In the PM model, $P(k|j)$ in (19) are fixed for each j such that $P(k|j) = 1$ for one class and $P(k|j) = 0$ for the other classes. We designed an MoE-PM classifier, where each class had two probability models $p(\boldsymbol{x}|j; \boldsymbol{\theta}_j)$. NB models were employed as the probability models in MoE-PM and MoE-PM. The joint probability models in MoE-GM and MoE-PM were trained by using EM-λ, because EM-λ provided better classification performance than the EM algorithm in our preliminary experiments.

TABLE 2
Classification Accuracies (in Percent) with Hybrid and MoE Classifiers Trained on N Labeled Samples with M Fixed Unlabeled Samples

N	N/M	Hybrid	MoE-PM	MoE-GM
16	.0036	84.1 (3.7)	**84.5** (5.3)	68.0 (19.8)
32	.0071	**89.4** (1.6)	85.0 (4.8)	88.3 (1.6)*
64	.014	**91.4** (0.8)	88.9 (2.5)*	88.2 (3.2)*
128	.028	**92.8** (0.7)	90.5 (1.4)*	90.1 (2.3)*
256	.057	**93.2** (0.9)	92.4 (1.0)	92.4 (1.0)
512	.11	**94.1** (0.4)	93.0 (1.4)*	93.2 (0.8)*
1024	.23	**94.7** (0.2)	93.8 (0.7)*	93.8 (0.7)*

(a)

N	N/M	Hybrid	MoE-PM	MoE-GM
8	.0032	**61.4** (5.9)	58.7 (7.4)	54.0 (13.2)
16	.0064	**66.8** (5.0)	66.2 (5.1)	64.1 (6.2)
32	.013	**72.7** (3.3)	69.1 (3.1)*	68.7 (3.7)*
64	.026	**76.2** (2.1)	74.2 (2.6)*	74.2 (2.6)*
128	.051	**79.3** (1.5)	76.6 (1.7)*	76.7 (1.5)*
256	.10	**81.3** (1.5)	77.5 (1.5)*	76.7 (0.9)*
512	.20	**83.1** (1.7)	78.9 (1.2)*	79.2 (1.5)*

(b)

N	N/M	Hybrid	MoE-PM	MoE-GM
10	.0040	**52.9** (13.2)	38.8 (8.6)*	32.8 (17.8)*
20	.0080	**64.3** (6.3)	53.2 (5.0)*	53.1 (3.7)*
40	.016	**70.0** (2.4)	57.8 (4.0)*	56.9 (4.4)*
80	.032	**73.1** (2.1)	63.3 (2.4)*	63.4 (2.5)*
160	.064	**75.9** (1.5)	68.2 (2.3)*	68.9 (2.3)*
320	.13	**78.4** (0.9)	74.5 (2.1)*	74.6 (2.1)*
640	.26	**81.2** (1.2)	77.6 (1.2)*	77.5 (1.2)*
1280	.51	**83.8** (1.1)	81.1 (1.2)*	80.7 (1.4)*

(c)

N	N/M	Hybrid	MoE-PM	MoE-GM
14	.0070	**65.8** (7.8)	47.9 (7.5)*	38.8 (16.4)*
28	.014	**72.6** (6.5)	58.2 (3.3)*	58.9 (5.0)*
56	.028	**77.0** (4.0)	66.0 (4.8)*	65.9 (4.7)*
112	.056	**82.9** (2.1)	73.4 (2.9)*	73.5 (2.9)*
224	.11	**85.5** (0.8)	80.8 (1.1)*	80.4 (2.0)*
448	.22	**87.0** (1.3)	81.1 (1.6)*	81.2 (1.7)*
896	.45	**88.4** (1.0)	84.8 (1.3)*	84.7 (1.4)*

(d)

(a) Reuters $(M = 4,500)$. *(b) WebKB* $(M = 2,500)$. *(c) 20news* $(M = 2,500)$. *(d) Cora* $(M = 2,000)$.

Table 2 shows the average classification accuracies for 10 experiments undertaken with our hybrid approach, MoE-PM, and MoE-GM. We ran the experiments using the 10 different sample sets described in Section 4.3.1. Each number in parentheses in the table denotes the standard deviation of the 10 experiments. N and M represent the number of labeled and unlabeled samples, respectively. Asterisks show that the difference between the average of our hybrid classifier and the classifiers used for comparison is significant $(p < 0.05)$ in the Wilcoxon test.

As shown in Table 2, our hybrid approach outperformed MoE-PM and MoE-GM. This result indicates that the good classification performance of our hybrid approach was not caused simply by adding different generative model parameters. In our hybrid approach, we introduce bias correction models to incorporate global data distribution by

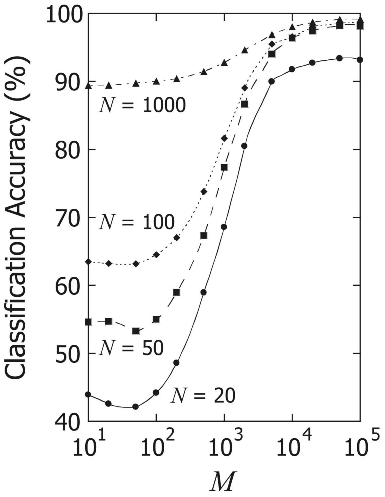


Fig. 5. Classification accuracies (in percent) with our proposed hybrid classifier trained on M unlabeled samples with N fixed labeled samples for an artificial data set.

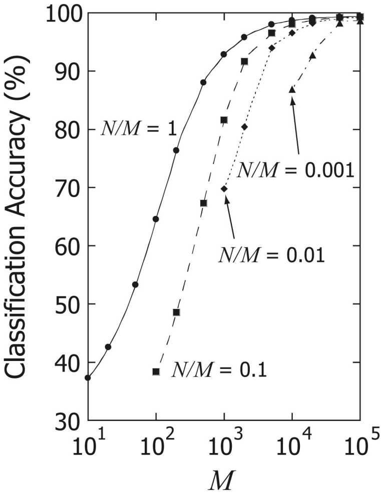


Fig. 6. Classification accuracies (in percent) with our proposed hybrid classifier trained on M unlabeled samples with a fixed ratio for the numbers of labeled and unlabeled samples N/M for an artificial data set.

using unlabeled samples. Then, the bias correction models are combined discriminatively with generative models. We think that this formulation for the bias correction models was effective in improving classification performance.

5 Experiments Using Different Unlabeled Sample Settings

We compared the performance of our hybrid classifier with those of generative and discriminative classifiers for four test collections and confirmed the usefulness of our hybrid approach experimentally. In this section, we show the effect of using a massive number of unlabeled samples on the performance of our hybrid classifier. We also show the performance of the hybrid classifier obtained by using noisy and biased unlabeled sample sets, where the distribution of unlabeled samples is different from that of labeled samples. These settings would be used in actual semisupervised classification tasks. For experiments using these settings, we employed a sparse artificial data set such as text.

5.1 Artificial Data Set

The artificial data set that we used consisted of 20 classes. Each data sample was generated by using one of the 20 NB models $\{p(\boldsymbol{x}|k;\boldsymbol{\theta}_k)\}_{k=1}^{20}$. The NB models were obtained by using all the articles included in the 20news data set. The vocabulary size of the NB models was 52,647. Words included in a data sample were selected independently from each other and with the probabilities provided by one of the NB models. An average of about 19 words was included in each data sample.

We selected five classes based on *comp.** to consider a five-class classification problem. Labeled and test samples were selected from data samples belonging to the five classes. Data samples belonging to the other classes were used to obtain a noisy unlabeled sample set whose distribution was different from that of labeled samples.

5.2 Effect of Massive Number of Unlabeled Samples

We examined the performance of our hybrid classifier trained by using various numbers of labeled and unlabeled samples to confirm the effect of using a massive number of unlabeled samples. For this examination, we used unlabeled samples collected from the same distribution as the labeled samples. In each experiment, labeled, unlabeled, and 10,000 test samples were selected randomly from samples whose true class labels were one of the five classes.

Fig. 5 shows the average classification accuracies of 10 experiments with our hybrid classifier trained by using M unlabeled samples with N fixed labeled samples. As shown in this figure, the classification accuracies obtained by using a massive number of unlabeled samples tended to be higher than those obtained by using a small number of unlabeled samples with a fixed number of labeled samples. When $N = 50, 100, \text{and } 1,000$, the classifier trained by using more than 10^4 unlabeled samples provided a classification accuracy of more than 98 percent.

We also show the classification accuracies at a constant ratio N/M as a function of M in Fig. 6. The classification performance of our hybrid classifier with a small N/M was much worse than that with a large N/M when there were a small number of unlabeled samples. By contrast, a high classification accuracy was obtained for a small N/M when there were a massive number of unlabeled samples. These results indicate the usefulness of a massive number of unlabeled samples for improving the classification performance when the unlabeled samples came from the same distribution as the labeled samples.

5.3 Effect of Noisy and Biased Unlabeled Sample Sets

We evaluated the performance of our hybrid classifier trained by using noisy and biased unlabeled sample sets, where the distribution of the unlabeled samples was different from that of the labeled samples. For the evaluation, we compared the performance of our hybrid classifier with four semisupervised classifiers based on NB/EM-λ, MLR/MER, MoE-PM, and MoE-GM used for the experiments described in Sections 4.3 and 4.4. The performance of these classifiers was examined for five-class classification problems in the artificial and real 20news data sets.

TABLE 3
Classification Accuracies (in Percent) with Semisupervised Classifiers Trained by Using an Unlabeled Sample Set Containing α Percent Noise Samples

α	N	Hybrid	NB/EM-λ	MoE-PM	MoE-GM	MLR/MER
0	20	91.9 (4.7)	**95.3** (3.6)	84.2 (6.5)*	93.0 (7.9)	49.6 (3.2)*
	100	97.2 (0.3)	**97.9** (0.2)*	95.5 (3.4)	94.4 (4.3)	66.9 (3.4)*
	1000	98.0 (0.2)	**98.3** (0.1)*	97.8 (0.6)	97.8 (0.6)	88.9 (0.4)*
50	20	72.9 (8.6)	69.0 (6.2)	72.9 (5.1)	**76.1** (5.9)	48.4 (2.8)*
	100	**91.0** (1.4)	89.6 (5.8)	87.7 (4.9)	87.3 (4.8)	65.3 (2.3)*
	1000	95.9 (0.4)	**96.3** (0.2)*	95.7 (0.7)	95.7 (0.7)	85.8 (0.5)*
90	20	**53.7** (5.9)	53.4 (4.8)	51.4 (5.1)	51.5 (4.7)	47.2 (2.1)*
	100	**73.1** (2.8)	70.3 (5.0)	67.8 (2.9)*	67.9 (3.2)*	64.7 (1.9)*
	1000	90.6 (0.4)	**91.6** (0.3)*	89.0 (1.4)*	89.1 (1.4)*	85.8 (0.5)*

(a)

α	N	Hybrid	NB/EM-λ	MoE-PM	MoE-GM	MLR/MER
0	20	**62.7** (4.5)	50.7 (4.3)*	52.0 (4.3)*	51.7 (6.8)*	45.4 (2.7)*
	100	**73.9** (1.9)	63.8 (4.4)*	64.6 (4.1)*	64.7 (4.1)*	64.8 (3.2)*
	1000	**83.3** (1.0)	78.3 (2.1)*	80.4 (1.4)*	80.0 (1.3)*	81.9 (1.2)*
50	20	**49.7** (3.1)	44.7 (4.6)*	44.5 (6.1)*	44.7 (4.9)*	44.9 (2.2)*
	100	**64.5** (2.8)	59.8 (3.8)*	62.1 (2.6)*	61.9 (2.7)*	63.0 (2.7)
	1000	**81.6** (1.1)	77.9 (2.1)*	78.3 (1.7)*	78.7 (1.1)*	80.8 (1.0)*
90	20	41.7 (3.2)	36.8 (4.5)*	39.1 (3.6)	38.6 (4.7)*	**43.7** (2.2)
	100	58.5 (2.9)	55.6 (3.9)	55.7 (3.9)	55.7 (3.9)	**62.5** (2.8)*
	1000	**81.1** (1.3)	76.3 (2.4)*	77.9 (1.0)*	77.7 (1.3)*	80.8 (0.9)

(b)

(a) Artificial Data Set $(M = 10{,}000)$. *(b) Real 20news Data Set* $(M = 2{,}500)$.

5.3.1 Noisy Unlabeled Sample Set

We examined the performance of our hybrid classifier and the four other semisupervised classifiers when they were trained by using noisy unlabeled sample sets, which contained samples unrelated to the target classes in the classification problems. The unrelated samples were collected from the fifteen classes that were not used for selecting labeled and test samples. We made noisy unlabeled sample sets by blending samples related and unrelated to the target classes. In each experiment, 10,000 and 1,000 test samples were collected randomly from the artificial and real 20news data sets, respectively, and labeled and unlabeled samples were selected randomly from the remaining samples.

Table 3 shows the average classification accuracies of 10 experiments with the five semisupervised classifiers trained by using N labeled samples and M unlabeled samples. α (in percent) represents the ratio of unrelated samples in an unlabeled sample set. Each number in parentheses in the table denotes the standard deviation of the 10 experiments. Asterisks show that the difference between the average classification accuracies of our hybrid classifier and the classifiers used for comparison is significant $(p < 0.05)$ in the Wilcoxon test.

Our hybrid approach outperformed NB/EM-λ, MoE-PM, and MoE-GM in all cases for the real 20news data set. The classification performance of our hybrid approach was not always worse than that of NB/EM-λ, MoE-PM, and MoE-GM when $\alpha = 50$ and 90 for the artificial data samples, where the classification performance of our hybrid approach was worse than NB/EM-λ when $\alpha = 0$. We confirmed that our hybrid approach was useful for obtaining a better classifier from noisy unlabeled sample sets, especially when the performance of our hybrid classifier trained on unlabeled sample sets not containing unrelated samples was better than that of classifiers based on NB/EM-λ, MoE-PM, and MoE-GM. The good classification performance of NB/EM-λ for the artificial data set would be reasonable because the artificial data samples were generated by using NB models and NB/EM-λ trained NB models to fit them into the artificial data samples.

Our hybrid approach did not outperform MLR/MER when $N = 20$ and 100 and $\alpha = 90$ for the real 20news data set. In these cases, we cannot expect to improve the classification performance greatly by using an unlabeled sample set because an unlabeled sample set contains a small number (250) of samples related to target classes. In supervised cases, MLR outperformed NB, as shown in Table 1c. These conditions would have caused the results whereby our hybrid approach employing NB models did not always outperform MLR/MER.

TABLE 4
Classification Accuracies (in Percent) with Classifiers Trained by Using Biased (B) and Unbiased (UB) Training Sets

Bias	N	Hybrid	NB/EM-λ	MoE-PM	MoE-GM	MLR/MER
UB	20	91.9 (4.7)	**95.3** (3.6)	84.2 (6.5)*	93.0 (7.9)	49.6 (3.2)*
	100	97.2 (0.3)	**97.9** (0.2)*	95.5 (3.4)	94.4 (4.3)	66.9 (3.4)*
	1000	98.0 (0.2)	**98.3** (0.1)*	97.8 (0.6)	97.8 (0.6)	88.9 (0.4)*
B	20	38.3 (10.3)	37.5 (17.5)	42.0 (10.1)	**44.0** (12.6)	30.3 (4.1)*
	100	41.1 (4.4)	35.7 (7.2)	**44.1** (6.4)	41.3 (6.1)	33.6 (0.6)*
	1000	64.2 (6.2)	61.9 (23.8)	**73.9** (5.6)*	73.0 (10.5)*	40.2 (1.1)*

(a)

Bias	N	Hybrid	NB/EM-λ	MoE-PM	MoE-GM	MLR/MER
UB	20	**62.7** (4.5)	50.7 (4.3)*	52.0 (4.3)*	51.7 (6.8)*	45.4 (2.7)*
	100	**73.9** (1.9)	63.8 (4.4)*	64.6 (4.1)*	64.7 (4.1)*	64.8 (3.2)*
	1000	**83.3** (1.0)	78.3 (2.1)*	80.4 (1.4)*	80.0 (1.3)*	81.9 (1.2)*
B	20	**40.5** (5.8)	37.0 (6.1)	35.1 (6.2)*	35.6 (6.7)	33.3 (3.7)*
	100	**52.9** (8.3)	43.3 (8.6)*	50.8 (5.1)	49.3 (6.6)*	39.8 (4.4)*
	1000	60.7 (4.0)	57.5 (8.5)	64.1 (6.2)	**64.3** (5.8)	52.6 (3.1)*

(b)

(a) Artificial Data Set $(M = 10{,}000)$. *(b) Real 20news Data Set* $(M = 2{,}500)$.

5.3.2 Biased Unlabeled Sample Set

We also evaluated the performance of our hybrid classifier and the four other semisupervised classifiers obtained by using biased unlabeled sample sets, whose feature distribution was different from that of labeled sample sets. The effect of labeled and unlabeled sample sets with different feature distributions on classification performance was also examined in [29]. For the evaluation, only samples whose true class labels were one of the five target classes were used as labeled, unlabeled, and test samples. In each experiment, 10,000 and 1,000 test samples were selected randomly from the artificial and real 20news data sets, respectively. To obtain biased training sets, the remaining samples, except for the test samples, were divided into two subsets by using a spherical K-means clustering algorithm [30], which was developed for dividing high-dimensional and sparse data such as text. Then, N labeled samples and M unlabeled samples were selected from the different subsets. We also obtained unbiased training sets by selecting labeled and unlabeled samples randomly from the remaining samples except for the test samples.

Table 4 shows the average classification accuracies of 10 experiments with the five semisupervised classifiers obtained by using the biased and unbiased training sample

TABLE 5
Class Distribution of Biased (B) and Unbiased (UB) Training Sets

Bias	N	Labeled Sample Set	Unlabeled Sample Set
UB	20	18.5: 21.0: 25.0: 18.0: 17.5	20.0: 20.2: 20.1: 19.8: 19.8
	100	18.4: 18.7: 21.3: 20.7: 20.9	
	1000	20.2: 19.8: 20.1: 19.3: 20.6	
B	20	33.0: 11.0: 16.0: 14.5: 25.5	7.1: 32.4: 27.1: 26.2: 7.2
	100	36.9: 4.2: 10.8: 12.1: 36.0	
	1000	38.1: 2.7: 10.2: 10.9: 38.0	

(a)

Bias	N	Labeled Sample Set	Unlabeled Sample Set
UB	20	21.5: 20.0: 19.0: 17.0: 22.5	19.9: 20.1: 19.9: 20.0: 20.0
	100	20.0: 21.1: 19.0: 17.0: 22.9	
	1000	19.9: 20.3: 20.5: 19.3: 20.0	
B	20	17.5: 22.0: 20.5: 22.0: 18.0	25.8: 20.6: 16.4: 17.1: 20.1
	100	10.7: 19.4: 24.6: 24.1: 21.2	
	1000	8.7: 19.0: 27.1: 24.8: 20.4	

(b)

(a) Artificial Data Set $(M = 10{,}000)$. *(b) Real 20news Data Set* $(M = 2{,}500)$.

TABLE 6
The Numbers of Vocabulary Words Included in Labeled and Unlabeled Sample Sets W_l and W_u Included in Both Labeled and Unlabeled Sample Sets $W_{l\wedge u}$

Bias	N	W_l	W_u	$W_{l\wedge u}$	$W_{l\wedge u}/W_l$
UB	20	332	18084	321	0.969
	100	1334		1282	0.961
	1000	6462		5946	0.920
B	20	313	15323	275	0.879
	100	1307		1065	0.816
	1000	5860		4051	0.693

(a)

Bias	N	W_l	W_u	$W_{l\wedge u}$	$W_{l\wedge u}/W_l$
UB	20	946	20047	862	0.926
	100	3510		3140	0.899
	1000	13542		10856	0.802
B	20	769	16208	695	0.904
	100	3185		2559	0.816
	1000	9820		7288	0.746

(b)

(a) Artificial Data Set $(M = 10{,}000)$. *(b) Real 20news Data Set* $(M = 2{,}500)$.

sets. Table 5 shows the average class distributions of the labeled and unlabeled sample sets used for the 10 experiments. Table 6 summarizes the numbers of vocabulary words included in the labeled and unlabeled sample sets W_l and W_u and included in both the labeled and unlabeled sample sets $W_{l\wedge u}$ to show the difference between the feature distributions of labeled and unlabeled samples. As shown in Table 6, the $W_{l\wedge u}/W_l$ values for biased training sets were smaller than those for unbiased training sets. This indicates that a feature space shared by labeled and unlabeled samples was smaller for the biased training sets.

When biased training sets were used, the classification performance of our hybrid approach, MoE-PM, and MoE-GM was better than that of NB/EM-λ for the artificial and real 20news data sets. Our hybrid approach, MoE-PM, and MoE-GM employed $2K$ NB models to construct K-class classifiers, whereas NB/EM-λ used K NB models. Using many NB models might be effective for fitting classifiers into labeled and unlabeled samples whose feature distributions are different.

Our hybrid approach outperformed MoE-PM and MoE-GM when using unbiased training sets for both data sets. However, when biased training sets were used, our hybrid approach provided worse classification performance than MoE-PM and MoE-GM for the artificial data set and when $N = 1{,}000$ for the real 20news data set. As shown in Table 6, the number of vocabulary words included in the biased labeled sample sets was smaller than that in the unbiased labeled sample sets. The discriminative training in our hybrid approach would have overfitted classifiers to labeled samples existing in a small part of the feature space.

6 Conclusion

We proposed a new approach to semisupervised classifier design based on a hybrid formed from the generative and discriminative approaches. The main idea is to introduce a bias correction model with different parameterization to correct the bias associated with a generative model trained on labeled samples.

In our experiments, we employed four actual data sets for text classification problems and confirmed that the use of a large number of unlabeled samples for training our hybrid classifier greatly improved the classification performance when the number of labeled samples was insufficiently large to obtain good classification performance. We compared our hybrid approach with conventional generative and discriminative approaches. Our approach greatly outperformed both these approaches when their classification performance was comparable. In other words, we can suggest that our hybrid classifier is useful when the discriminative classifier performs similarly to the generative classifier. We also confirmed that our hybrid approach had an advantage over the generative and discriminative approaches in terms of processing time. Moreover, we examined the performance of our hybrid classifier when the labeled and unlabeled samples had different distributions.

Future work will involve applying our hybrid approach to other data, where different generative models are employed, to confirm that the hybrid generative and discriminative approach is useful for designing semisupervised classifiers for various types of data.

Appendix A

Derivation of Q-Function for Parameter Estimation of Bias Correction Model

We derive the Q-function $Q(\Psi^{(t+1)}, \Psi^{(t)}|\Gamma)$ shown in (12) from $G(\Psi|\Gamma)$ shown in (11). We define

$$z_{mk}(\Psi) \equiv \frac{g_k(\boldsymbol{x}_m; \boldsymbol{\psi}_k)}{\sum_{k'=1}^{K} g_{k'}(\boldsymbol{x}_m; \boldsymbol{\psi}_{k'})}, \tag{20}$$

$$G'(\Psi|\Gamma) \equiv \sum_{m=1}^{M} \log \sum_{k=1}^{K} g_k(\boldsymbol{x}_m; \boldsymbol{\psi}_k). \tag{21}$$

Then, the difference in $G'(\Psi|\Gamma)$ between the $(t+1)\mathrm{th}$ and $(t)\mathrm{th}$ steps is written as

$$\begin{aligned}
&G'\left(\Psi^{(t+1)}|\Gamma\right) - G'\left(\Psi^{(t)}|\Gamma\right)\\
&= \sum_{m=1}^{M} \log \frac{\sum_{k=1}^{K} g_k\left(\boldsymbol{x}_m; \Psi^{(t+1)}\right)}{\sum_{k=1}^{K} g_k\left(\boldsymbol{x}_m; \Psi^{(t)}\right)}\\
&= \sum_{m=1}^{M}\sum_{k'=1}^{K}\left[z_{mk'}\left(\Psi^{(t)}\right) \log\left\{ \frac{\sum_{k=1}^{K} g_k\left(\boldsymbol{x}_m; \boldsymbol{\psi}_k^{(t+1)}\right)}{\sum_{k=1}^{K} g_k\left(\boldsymbol{x}_m; \boldsymbol{\psi}_k^{(t)}\right)}\right.\right.\\
&\quad \left.\left.\times \frac{g_{k'}\left(\boldsymbol{x}_m; \boldsymbol{\psi}_{k'}^{(t+1)}\right)}{g_{k'}\left(\boldsymbol{x}_m; \boldsymbol{\psi}_{k'}^{(t+1)}\right)} \frac{g_{k'}\left(\boldsymbol{x}_m; \boldsymbol{\psi}_{k'}^{(t)}\right)}{g_{k'}\left(\boldsymbol{x}_m; \boldsymbol{\psi}_{k'}^{(t)}\right)} \right\}\right]\\
&= \sum_{m=1}^{M}\sum_{k'=1}^{K}\left[z_{mk'}\left(\Psi^{(t)}\right)\right.\\
&\quad \left.\times \log\left\{ \frac{g_{k'}\left(\boldsymbol{x}_m; \boldsymbol{\psi}_k^{(t+1)}\right)}{g_{k'}\left(\boldsymbol{x}_m; \boldsymbol{\psi}_k^{(t)}\right)} \frac{z_{mk'}\left(\Psi^{(t)}\right)}{z_{mk'}\left(\Psi^{(t+1)}\right)} \right\}\right]\\
&= \sum_{m=1}^{M}\sum_{k=1}^{K} z_{mk}\left(\Psi^{(t)}\right) \log \frac{p\left(\boldsymbol{x}_m|k; \boldsymbol{\psi}_k^{(t+1)}\right)^{\gamma_2}}{p\left(\boldsymbol{x}_m|k; \boldsymbol{\psi}_k^{(t)}\right)^{\gamma_2}}\\
&\quad + \sum_{m=1}^{M}\sum_{k=1}^{K} z_{mk}\left(\Psi^{(t)}\right) \log \frac{z_{mk}\left(\Psi^{(t)}\right)}{z_{mk}\left(\Psi^{(t+1)}\right)}.
\end{aligned} \tag{22}$$

Here, the second term in (22) is the sum of Kullback-Leibler divergence of $z_{mk}(\Psi^{(t)})$ and $z_{mk}(\Psi^{(t+1)})$

$$\sum_{k=1}^{K} z_{mk}\left(\Psi^{(t)}\right) \log \frac{z_{mk}\left(\Psi^{(t)}\right)}{z_{mk}\left(\Psi^{(t+1)}\right)} \geq 0. \tag{23}$$

Defining $Q(\Psi^{(t+1)}, \Psi^{(t)}|\Gamma)$ such as (12) by using $z_{mk}(\Psi) = R(k|\boldsymbol{x}; \hat{\Theta}, \Psi, \Gamma)$ and $G(\Psi|\Gamma) = G'(\Psi|\Gamma) + \log p(\Psi)$, we can obtain the inequality

$$\begin{aligned}
&G\left(\Psi^{(t+1)}|\Gamma\right) - G\left(\Psi^{(t)}|\Gamma\right)\\
&\geq Q\left(\Psi^{(t+1)}, \Psi^{(t)}|\Gamma\right) - Q\left(\Psi^{(t)}, \Psi^{(t)}|\Gamma\right).
\end{aligned} \tag{24}$$

The inequality shows that $Q(\Psi^{(t+1)}, \Psi^{(t)}|\Gamma) - Q(\Psi^{(t)}, \Psi^{(t)}|\Gamma)$ provides the lower bound of the improvement of $G(\Psi|\Gamma)$ by the update of Ψ. Therefore, by computing $\Psi^{(t+1)}$ to maximize $Q(\Psi^{(t+1)}, \Psi^{(t)}|\Gamma)$, we can improve $G(\Psi|\Gamma)$ in the $(t+1)$th step. By iteratively performing this update, we can obtain an estimate of Ψ that provides the local maximum of $G(\Psi|\Gamma)$ around an initialized value of Ψ.

Appendix B

Hyperparameter Tuning Procedure

We explain the procedure for tuning hyperparameter ξ_k by using a leave-one-out cross validation and the EM algorithm, as mentioned in Section 3.4. According to a MAP estimation, using training samples except $\boldsymbol{x}_{n_k}$, we obtain $\hat{\theta}_{ki}^{(-n_k)}$ in (17) such as

$$\hat{\theta}_{ki}^{(-n_k)} = \frac{\sum_{n_{k'}=1}^{N_k} x_{n_{k'}i} - x_{n_k i} + \xi_k - 1}{N_k - 1 + V(\xi_k - 1)}. \tag{25}$$

As with estimates of parameters smoothed by Lidstone's law (cf., [31]), we can express $\hat{\theta}_{ki}^{(-n_k)}$ by

$$\hat{\theta}_{ki}^{(-n_k)} = \beta\phi_i^{(-n_k)} + (1-\beta)\frac{1}{V}, \tag{26}$$

where

$$\beta = \frac{N_k - 1}{N_k - 1 + V(\xi_k - 1)}, \tag{27}$$

$$\phi_i^{(-n_k)} = \frac{\sum_{n_{k'}=1}^{N_k} x_{n_{k'}i} - x_{n_k i}}{N_k - 1} \geq 0. \tag{28}$$

Here, $0 \leq \beta < 1$, and $\sum_{i=1}^{V} \phi_i^{(-n_k)} = 1$. Therefore, we can view $\hat{\theta}_{ki}^{(-n_k)}$ as a linear interpolation between $\phi_i^{(-n_k)}$ and $1/V$. Since β is independent of the training sample $\boldsymbol{x}_{n_k}$, we can regard $L(\xi_k)$ shown in (17) as a function of β

$$L(\beta) = \sum_{n_k=1}^{N_k}\sum_{i=1}^{V} x_{n_k i} \log\left\{\beta\phi_i^{(-n_k)} + (1-\beta)\frac{1}{V}\right\}. \tag{29}$$

We can use the EM algorithm for estimating β to maximize $L(\beta)$. In this estimation, global convergence is guaranteed, since $L(\beta)$ is a concave function. Such an estimation of the interpolation weight β with cross validation was also employed in *deleted interpolation* [32]. Using the estimate of β and (27), we obtain ξ_k to maximize $L(\xi_k)$ shown in (17).

Acknowledgments

Dr. Saito's work on this paper was done while he was with NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan.

References

[1] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Text Classification from Labeled and Unlabeled Documents Using EM," *Machine Learning,* vol. 39, pp. 103-134, 2000.

[2] Y. Grandvalet and Y. Bengio, "Semi-Supervised Learning by Entropy Minimization," *Advances in Neural Information Processing Systems 17,* MIT Press, pp. 529-536, 2005.

[3] M. Szummer and T. Jaakkola, "Kernel Expansions with Unlabeled Examples," *Advances in Neural Information Processing Systems 13,* MIT Press, pp. 626-632, 2001.

[4] M. Inoue and N. Ueda, "Exploitation of Unlabeled Sequences in Hidden Markov Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 12, pp. 1570-1581, Dec. 2003.

[5] M.R. Amini and P. Gallinari, "Semi-Supervised Logistic Regression," *Proc. 15th European Conf. Artificial Intelligence,* pp. 390-394, 2002.

[6] T. Joachims, "Transductive Inference for Text Classification Using Support Vector Machines," *Proc. 16th Int'l Conf. Machine Learning,* pp. 200-209, 1999.

[7] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," *Proc. 11th Ann. Conf. Computational Learning Theory,* vol. 11, 1998.

[8] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," *Proc. 20th Int'l Conf. Machine Learning,* pp. 912-919, 2003.

[9] M. Seeger, "Learning with Labeled and Unlabeled Data," technical report, Univ. of Edinburgh, 2001.

[10] A.Y. Ng and M.I. Jordan, "On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes," *Advances in Neural Information Processing Systems 14,* pp. 841-848, MIT Press, 2002.

[11] S. Tong and D. Koller, "Restricted Bayes Optimal Classifiers," *Proc. 17th Nat'l Conf. Artificial Intelligence,* pp. 658-664, 2000.

[12] R. Raina, Y. Shen, A.Y. Ng, and A. McCallum, "Classification with Hybrid Generative/Discriminative Models," *Advances in Neural Information Processing Systems 16,* MIT Press, 2004.

[13] A.L. Berger, S.A. Della Pietra, and V.J. Della Pietra, "A Maximum Entropy Approach to Natural Language Processing," *Computational Linguistics,* vol. 22, no. 1, pp. 39-71, 1996.

[14] A. Fujino, N. Ueda, and K. Saito, "A Hybrid Generative/Discriminative Approach to Semi-Supervised Classifier Design," *Proc. 20th Nat'l Conf. Artificial Intelligence,* pp. 764-769, 2005.

[15] A. Fujino, N. Ueda, and K. Saito, "Semi-Supervised Learning on Hybrid Generative/Discriminative Models," *Information Technology Letters,* vol. 4, pp. 161-164, 2005, in Japanese.

[16] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. B,* vol. 39, pp. 1-38, 1977.

[17] F.G. Cozman and I. Cohen, "Unlabeled Data Can Degrade Classification Performance of Generative Classifiers," *Proc. 15th Int'l Florida Artificial Intelligence Research Soc. Conf.,* pp. 327-331, 2002.

[18] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer-Verlag, 2001.

[19] K. Nigam, J. Lafferty, and A. McCallum, "Using Maximum Entropy for Text Classification," *Proc. Int'l Joint Conf. Artificial Intelligence Workshop Machine Learning for Information Filtering,* pp. 61-67, 1999.

[20] S.F. Chen and R. Rosenfeld, "A Gaussian Prior for Smoothing Maximum Entropy Models," technical report, Carnegie Mellon Univ., 1999.

[21] D.C. Liu and J. Nocedal, "On the Limited Memory BFGS Method for Large Scale Optimization," *Math. Programming B,* vol. 45, no. 3, pp. 503-528, 1989.

[22] A. Fujino, N. Ueda, and K. Saito, "A Hybrid Generative/Discriminative Approach to Text Classification with Additional Information," *Information Processing and Management,* vol. 43, pp. 379-392, 2007.

[23] Y. Yang and X. Liu, "A Re-Examination of Text Categorization Methods," *Proc. 22nd ACM Int'l Conf. Research and Development in Information Retrieval,* pp. 42-49, 1999.

[24] G. Salton and M.J. McGill, *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[25] G. Forman, "An Extensive Empirical Study of Feature Selection Metrics for Text Classification," *J. Machine Learning Research,* vol. 3, pp. 1289-1305, 2003.

[26] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter, "On Feature Distributional Clustering for Text Classification," *Proc. 24th ACM Int'l Conf. Research and Development in Information Retrieval,* pp. 146-153, 2001.

[27] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *J. Machine Learning Research,* vol. 7, pp. 1-30, 2006.

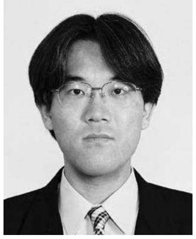
[28] D.J. Miller and H.S. Uyar, "A Mixture of Experts Classifier with Learning Based on Both Labelled and Unlabelled Data," *Advances in Neural Information Processing Systems 9,* pp. 571-577, MIT Press, 1997.

[29] N.V. Chawla and G. Karakoulas, "Learning from Labeled and Unlabeled Data: An Empirical Study across Techniques and Domains," *J. Artificial Intelligence Research,* vol. 23, pp. 331-366, 2005.

[30] I.S. Dhillon and D.S. Modha, "Concept Decompositions for Large Sparse Text Data Using Clustering," *Machine Learning,* vol. 42, pp. 143-175, 2001.

[31] C.D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing.* The MIT Press, 1999.

[32] F. Jelinek and R. Mercer, "Interpolated Estimation of Markov Source Parameters from Sparse Data," *Pattern Recognition in Practice,* E.S. Gelsema and L.N. Kanal, eds., pp. 381-402, North Holland Publishing, 1980.



Akinori Fujino received the BS and MS degrees in precision engineering from Kyoto University, Kyoto, Japan, in 1995 and 1997, respectively. In 1997, he joined the Nippon Telegraph and Telephone Corp. (NTT) Basic Research Laboratories, Kanagawa, Japan. In 2003, he joined the NTT Communication Science Laboratories, Kyoto. He is a research scientist at the Innovative Communication Laboratory. His current research interests are machine learning and information extraction from complex data. He is a member of the Institute of Electronics, Information, and Communication Engineers (IEICE), the Information Processing Society of Japan (IPSJ), and the IEEE.

Naonori Ueda received the BS, MS, and PhD degrees in communication engineering from Osaka University, Osaka, Japan, in 1982, 1984, and 1992, respectively. In 1984, he joined the Nippon Telegraph and Telephone Corp. (NTT) Electrical Communication Laboratories, Kanagawa, Japan. In 1991, he joined the NTT Communication Science Laboratories, Kyoto, as a senior research scientist. He is an executive manager of the Innovative Communication Laboratory. From 1993 to 1994, he was a visiting scholar at Purdue University, West Lafayette, Ind. His current research interests are statistical machine learning and its application to pattern recognition and data mining. He is a member of the Institute of Electronics, Information, and Communication Engineers (IEICE), the Information Processing Society of Japan (IPSJ), the Japanese Neural Network Society (JNNS), and the IEEE.

Kazumi Saito received the BS degree in mathematics from Keio University, Kanagawa, Japan, in 1985 and the PhD degree in engineering from the University of Tokyo, Tokyo, Japan, in 1998. In 1985, he joined the Nippon Telegraph and Telephone Corp. (NTT) Electrical Communication Laboratories, Kanagawa, Japan. In 1991, he joined the NTT Communication Science Laboratories, Kyoto. In 2007, he joined the University of Shizuoka, Shizuoka, Japan. He is a professor in the School of Administration and Informatics. From 1991 to 1992, he was a visiting scholar at the University of Ottawa, Ontario, Canada. His current research interests are machine learning and statistical analysis of complex networks. He is a member of the Institute of Electronics, Information, and Communication Engineers (IEICE), the Information Processing Society of Japan (IPSJ), the Japanese Society of Artificial Intelligence (JSAI), and the Japanese Neural Network Society (JNNS).