Semi-supervised Learning for Multi-component Data Classification

Akinori Fujino, Naonori Ueda, and **Kazumi Saito**
NTT Communication Science Laboratories
NTT Corporation
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan 619-0237
{a.fujino,ueda,saito}@cslab.kecl.ntt.co.jp

Abstract

This paper presents a method for designing a semi-supervised classifier for multi-component data such as web pages consisting of text and link information. The proposed method is based on a hybrid of generative and discriminative approaches to take advantage of both approaches. With our hybrid approach, for each component, we consider an individual generative model trained on labeled samples and a model introduced to reduce the effect of the bias that results when there are few labeled samples. Then, we construct a hybrid classifier by combining all the models based on the maximum entropy principle. In our experimental results using three test collections such as web pages and technical papers, we confirmed that our hybrid approach was effective in improving the generalization performance of multi-component data classification.

1 Introduction

Data samples such as web pages and multimodal data usually contain main and additional information. For example, web pages consist of main text and additional components such as hyperlinks and anchor text. Although the main content plays an important role when designing a classifier, additional content may contain substantial information for classification. Recently, classifiers have been developed that deal with *multi-component* data such as web pages [Chakrabarti *et al.*, 1998; Cohn and Hofmann, 2001; Sun *et al.*, 2002; Lu and Getoor, 2003], technical papers containing text and citations [Cohn and Hofmann, 2001; Lu and Getoor, 2003], and music data with text titles [Brochu and Freitas, 2003].

In supervised learning cases, existing probabilistic approaches to classifier design for arbitrary multi-component data are generative, discriminative, and a hybrid of the two. Generative classifiers learn the joint probability model, $p(\boldsymbol{x}, y)$, of feature vector $\boldsymbol{x}$ and class label y, compute $P(y|\boldsymbol{x})$ by using the Bayes rule, and then take the most probable class label y. To deal with multiple heterogeneous components, under the assumption of the class conditional independence of all components, the class conditional probability density $p(\boldsymbol{x}^j|y)$ for the jth component $\boldsymbol{x}^j$ is individually modeled [Brochu and Freitas, 2003]. By contrast, discriminative classifiers directly model class posterior probability $P(y|\boldsymbol{x})$ and learn mapping from $\boldsymbol{x}$ to y. Multinomial logistic regression [Hastie *et al.*, 2001] can be used for this purpose.

It has been shown that discriminative classifiers often achieve better performance than generative classifiers, but that the latter often provide better generalization performance than the former when trained by few labeled samples [Ng and Jordan, 2002]. Therefore, hybrid classifiers have been proposed to take advantage of the generative and discriminative approaches [Raina *et al.*, 2004]. To construct hybrid classifiers, the generative model $p(\boldsymbol{x}^j|y)$ of the jth component is first designed individually, and all the component models are combined with *weight* determined on the basis of a discriminative approach. It has been shown experimentally that the hybrid classifier performed better than pure generative and discriminative classifiers [Raina *et al.*, 2004].

On the other hand, a large number of labeled samples are often required if we wish to obtain better classifiers with generalization ability. 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 field of machine learning, and *semi-supervised* learning algorithms that use both labeled and unlabeled samples for training classifiers have been developed (cf. [Joachims, 1999; Nigam *et al.*, 2000; Grandvalet and Bengio, 2005; Fujino *et al.*, 2005b], see [Seeger, 2001] for a comprehensive survey).

In this paper, we focus on Semi-Supervised Learning for Multi-Component data classification (SSL-MC) based on probabilistic approach and present a *hybrid* classifier for SSL-MC problems. The hybrid classifier is constructed by extending our previous work for semi-supervised learning [Fujino *et al.*, 2005b] to deal with multiple components. In our formulation, an individual generative model for each component is designed and trained on labeled samples. When there are few labeled samples, the class boundary provided by the trained generative models is often far from being the most appropriate one. Namely, the trained generative models often have a high bias as a result of there being few labeled samples. To mitigate the effect of the bias on classification performance, for each component, we introduce a *bias correction model*. Then, by discriminatively combining these models based on the *maximum entropy principle* [Berger *et* *al.*, 1996], we obtain our hybrid classifier. The bias correction models are trained by using many unlabeled samples to incorporate global data distribution into the classifier design.

We can consider some straightforward applications of the conventional generative and discriminative semi-supervised learning algorithms [Nigam *et al.*, 2000; Grandvalet and Bengio, 2005] to train the above-mentioned classifiers for multi-component data. Using three test collections such as web pages and technical papers, we show experimentally that our proposed method is effective in improving the generalization performance of multi-component data classification especially when the generative and discriminative classifiers provide similar performance.

2 Straightforward Approaches to Semi-supervised Learning

2.1 Multi-component Data Classification

In multi-class (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. Here, each feature vector consists of J separate components as $\boldsymbol{x} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^j, \ldots, \boldsymbol{x}^J\}$. In semi-supervised 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 $D_u = \{\boldsymbol{x}_m\}_{m=1}^M$. 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 consider straightforward applications of conventional semi-supervised learning algorithms [Nigam *et al.*, 2000; Grandvalet and Bengio, 2005] with a view to incorporating labeled and unlabeled samples into generative, discriminative, and hybrid classifiers proposed for multi-component data.

2.2 Generative Classifiers

Generative classifiers model a joint probability density $p(\boldsymbol{x}, y)$. However, such direct modeling is hard for arbitrary components that consist of completely different types of media. To deal with *multi-component* data in the generative approach, under the assumption that all components have *class conditional independence*, the joint probability model can be expressed as $p(\boldsymbol{x}, k; \Theta) = P(k) \prod_{j=1}^J p(\boldsymbol{x}^j | k; \boldsymbol{\theta}_k^j)$ (cf. [Brochu and Freitas, 2003]), where $\boldsymbol{\theta}_k^j$ is the jth component model parameter for the kth class and $\Theta = \{\boldsymbol{\theta}_k^j\}_{j,k}$ is a set of model parameters. Note that the class conditional probability model $p(\boldsymbol{x}^j | k; \boldsymbol{\theta}_k^j)$ should be selected according to the features of the jth component. The class posteriors for all classes are computed according to the Bayes rule, such as

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

where $\hat{\Theta} = \{\hat{\boldsymbol{\theta}}_k^j\}_{j,k}$ is a parameter estimate set. The class label y of $\boldsymbol{x}$ is determined as the k that maximizes $P(k|\boldsymbol{x}; \hat{\Theta})$.

For the *semi-supervised* learning of the generative classifiers, unlabeled samples are dealt with as a missing class label problem, and are incorporated in a mixture of joint probability models [Nigam *et al.*, 2000]. That is, $\boldsymbol{x}_m \in D_u$ is drawn from the marginal generative distribution $p(\boldsymbol{x}; \Theta) = \sum_{k=1}^K p(\boldsymbol{x}, k; \Theta)$. Model parameter set Θ is computed by maximizing the posterior $p(\Theta | D)$ (MAP estimation). According to the Bayes rule, $p(\Theta|D) \propto p(D|\Theta)p(\Theta)$, we can provide the objective function of model parameter estimation such as

$$\begin{aligned} F(\Theta) &= \sum_{n=1}^N \log P(y_n) \prod_{j=1}^J p(\boldsymbol{x}_n^j | y_n; \boldsymbol{\theta}_{y_n}^j) \\ &+ \sum_{m=1}^M \log \sum_{k=1}^K P(k) \prod_{j=1}^J p(\boldsymbol{x}_m^j | k; \boldsymbol{\theta}_k^j) \\ &+ \log p(\Theta). \end{aligned} \quad (2)$$

Here, $p(\Theta)$ is a prior over Θ. We can obtain the Θ value that maximizes $F(\Theta)$ using the Expectation-Maximization (EM) algorithm [Dempster *et al.*, 1977].

The estimation of Θ is affected by the number of unlabeled samples used with labeled samples. In other words, when $N \ll M$, model parameter Θ is estimated as almost unsupervised clustering. Then, 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. To mitigate the problem, a weighting parameter $\lambda \in [0, 1]$ that reduces the contribution of the unlabeled samples to the parameter estimation was introduced (EM-λ [Nigam *et al.*, 2000]). The value of λ is determined by cross-validation so that the leave-one-out labeled samples are, as far as possible, correctly classified.

2.3 Discriminative Classifiers

Discriminative classifiers directly model class posterior probabilities $P(y|\boldsymbol{x})$ for all classes. We can design a discriminative model $P(y|\boldsymbol{x})$ without the separation of components. In Multinomial Logistic Regression (MLR) [Hastie *et al.*, 2001], the class posterior probabilities are modeled as

$$P(y = k | \boldsymbol{x}; W) = \frac{\exp(\boldsymbol{w}_k \cdot \boldsymbol{x})}{\sum_{k'=1}^K \exp(\boldsymbol{w}_{k'} \cdot \boldsymbol{x})}, \quad (3)$$

where $W = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k, \ldots, \boldsymbol{w}_K\}$ is a set of unknown model parameters. $\boldsymbol{w}_k \cdot \boldsymbol{x}$ represents the inner product of $\boldsymbol{w}_k$ and $\boldsymbol{x}$.

Certain assumptions are required if we are to incorporate unlabeled samples into discriminative classifiers, because $p(x)$ is not modeled in the classifiers. A *Minimum Entropy Regularizer* (MER) was introduced as one approach to the semi-supervised learning of discriminative classifiers [Grandvalet and Bengio, 2005]. This approach is based on the empirical knowledge that classes should be well separated to take advantage of unlabeled samples because the asymptotic information content of unlabeled samples decreases as classes overlap. The 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 with the regularizer:

$$F(W) = \sum_{n=1}^N \log P(y_n | \boldsymbol{x}_n; W)$$

$$+ \ \lambda \sum_{m=1}^{M} \sum_{k=1}^{K} P(k|\boldsymbol{x}_m; W) \log P(k|\boldsymbol{x}_m; W)$$
$$+ \ \log p(W). \tag{4}$$

Here, λ is a weighting parameter, and $p(W)$ is a prior over W. A Gaussian prior [Chen and Rosenfeld, 1999] is often employed as $p(W)$.

2.4 Hybrid Classifiers

Hybrid classifiers learn an individual class conditional probability model $p(\boldsymbol{x}^j|y;\boldsymbol{\theta}^j_y)$ for the jth component and directly model the class posterior probability by using the trained component models [Raina *et al.*, 2004; Fujino *et al.*, 2005a]. 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. For the hybrid classifiers, the class posteriors are provided as

$$R(y = k|\boldsymbol{x}; \hat{\Theta}, \Gamma)$$
$$= \frac{e^{\mu_k} \prod_{j=1}^{J} p(\boldsymbol{x}^j|k;\hat{\boldsymbol{\theta}}^j_k)^{\gamma_j}}{\sum_{k'=1}^{K} e^{\mu_{k'}} \prod_{j=1}^{J} p(\boldsymbol{x}^j|k';\hat{\boldsymbol{\theta}}^j_{k'})^{\gamma_j}}, \tag{5}$$

where $\hat{\Theta} = \{\hat{\boldsymbol{\theta}}_{j,k}\}_{j,k}$ is a parameter estimate set of component generative models $\{p(\boldsymbol{x}^j|k;\boldsymbol{\theta}^j_k)\}_{j,k}$, and $\Gamma = \{\{\gamma_j\}_{j=1}^{J}, \{\mu_k\}_{k=1}^{K}\}$ is a parameter set that provides the combination weights of components and class biases. In supervised learning cases, Γ is estimated to maximize the cross-validation conditional log-likelihood of labeled samples.

To incorporate unlabeled samples in the hybrid classifiers, we can consider a simple approach in which we train the component generative models using EM-λ as mentioned in Section 2.2. With this approach, we incorporate unlabeled samples into the component generative models and then discriminatively combine these models. For convenience, we call this approach *Cascade Hybrid* (CH).

3 Proposed Method

As mentioned in the introduction, we propose a semi-supervised hybrid classifier for multi-component data, where bias correction models are introduced to use unlabeled samples effectively. For convenience, we call this classifier *Hybrid classifier with Bias Correction Models* (H-BCM classifier). In this section, we present the formulation of the H-BCM classifier and its parameter estimation method.

3.1 Component Generative Models and Bias Correction

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

$$F_1(\Theta) = \sum_{j=1}^{J} \left\{ \sum_{n=1}^{N} \log p(\boldsymbol{x}^j_n|y_n;\boldsymbol{\theta}^j_{y_n}) + \sum_{k=1}^{K} \log p(\boldsymbol{\theta}^j_k) \right\}. \tag{6}$$

Here, $p(\boldsymbol{\theta}^j_k)$ is a prior over $\boldsymbol{\theta}^j_k$.

When there are few labeled samples, the classifier obtained by using the trained component generative models often provides a class boundary that is far from the most appropriate one. Namely, the trained component generative models often have a high bias. To obtain a classifier with a smaller bias, we newly introduce another class conditional generative model per component, called *bias correction model*. The generative and bias correction models for each component belong to the *same* model family, but the set of parameters $\Psi = \{\boldsymbol{\psi}^j_k\}_{j,k}$ of the bias correction models is different from Θ. We construct our hybrid classifier by combining the trained component generative models with the bias correction models to mitigate the effect of the bias.

3.2 Discriminative Class Posterior Design

We define our hybrid classifier by using the class posterior probability distribution derived from a discriminative combination of the component generative and bias correction models. The combination is provided 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, $\log p(\boldsymbol{x}^j|k;\hat{\boldsymbol{\theta}}^j_k)$, with respect to the target distribution $R(k|\boldsymbol{x})$ is equal to the expectation of the 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

$$\sum_{\boldsymbol{x},k} \tilde{p}(\boldsymbol{x}, k) \log p(\boldsymbol{x}^j|k;\hat{\boldsymbol{\theta}}^j_k)$$
$$= \sum_{\boldsymbol{x},k} \tilde{p}(\boldsymbol{x}) R(k|\boldsymbol{x}) \log p(\boldsymbol{x}^j|k;\hat{\boldsymbol{\theta}}^j_k), \ \forall j, \tag{7}$$

where $\tilde{p}(\boldsymbol{x}) = \sum_{n=1}^{N} \delta(\boldsymbol{x} - \boldsymbol{x}_n)/N$ is the empirical distribution of $\boldsymbol{x}$. The equation of the constraint for $\log p(\boldsymbol{x}^j|k;\boldsymbol{\psi}^j_k)$ can be represented in the same form as Eq. (7). We also restrict $R(k|\boldsymbol{x})$ so that it has the same class probability as seen in the labeled samples, such that

$$\sum_{\boldsymbol{x}} \tilde{p}(\boldsymbol{x}, k) = \sum_{\boldsymbol{x}} \tilde{p}(\boldsymbol{x}) R(k|\boldsymbol{x}), \ \forall k. \tag{8}$$

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:

$$R(y = k|\boldsymbol{x}; \hat{\Theta}, \Psi, \Gamma)$$
$$= \frac{e^{\mu_k} \prod_{j=1}^{J} p(\boldsymbol{x}^j|k;\hat{\boldsymbol{\theta}}^j_k)^{\gamma_{1j}} p(\boldsymbol{x}^j|k;\boldsymbol{\psi}^j_k)^{\gamma_{2j}}}{\sum_{k'=1}^{K} e^{\mu_{k'}} \prod_{j=1}^{J} p(\boldsymbol{x}^j|k';\hat{\boldsymbol{\theta}}^j_{k'})^{\gamma_{1j}} p(\boldsymbol{x}^j|k';\boldsymbol{\psi}^j_{k'})^{\gamma_{2j}}}, \tag{9}$$

where $\Gamma = \{\{\gamma_{1j}, \gamma_{2j}\}_{j=1}^{J}, \{\mu_k\}_{k=1}^{K}\}$ is a set of Lagrange multipliers. γ_{1j} and γ_{2j} represent the combination weights of the generative and bias correction models for the jth component, and μ_k is the bias parameter for the kth class. The distribution $R(k|\boldsymbol{x}, \hat{\Theta}, \Psi, \Gamma)$ gives us the formulation of the discriminative classifier, which consists of the trained component generative models and the bias correction models.

According to the ME principle, the solution of Γ in Eq. (9) is the same as the Γ that maximizes the log-likelihood for $R(k|\boldsymbol{x}; \hat{\Theta}, \Psi, \Gamma)$ of labeled samples $(\boldsymbol{x}_n, y_n) \in D_l$ [Berger *et al.*, 1996; Nigam *et al.*, 1999]. However, D_l is also used to estimate Θ. Using the same labeled samples for both Γ and Θ may lead to a bias estimation of Γ. Thus, a leave-one-out cross-validation of the labeled samples is used to estimate Γ [Raina *et al.*, 2004]. Let $\hat{\Theta}^{(-n)}$ be a generative model parameter set estimated by using all the labeled samples except $(\boldsymbol{x}_n, y_n)$. The objective function of Γ then becomes

$$F_2(\Gamma|\Psi) = \sum_{n=1}^{N} \log R(y_n|\boldsymbol{x}_n; \hat{\Theta}^{(-n)}, \Psi, \Gamma) + \log p(\Gamma), \quad (10)$$

where $p(\Gamma)$ is a prior over Γ. We used the Gaussian prior [Chen and Rosenfeld, 1999] as $p(\Gamma) \propto \prod_k \exp(-\mu_k^2/2\rho^2) \prod_{l,j} \exp\{-(\gamma_{lj}-1)^2/2\sigma^2\}$. Global convergence is guaranteed when Γ is estimated with fixed $\hat{\Theta}^{(-n)}$ and Ψ, since $F_2(\Gamma|\Psi)$ is an upper convex function of Γ.

3.3 Training Bias Correction Models

In our formulation, the parameter set Ψ of the bias correction models is trained with unlabeled samples to reduce the bias that results when there are few labeled samples. According to $R(k|\boldsymbol{x}; \hat{\Theta}, \Psi, \Gamma)$ as shown in Eq. (9), the class label y of a feature vector $\boldsymbol{x}$ is determined as the k that maximizes the discriminative function:

$$g_k(\boldsymbol{x}; \Psi) \;=\; e^{\mu_k} \prod_{j=1}^{J} p(\boldsymbol{x}^j|k; \hat{\boldsymbol{\theta}}_k^j)^{\gamma_1} p(\boldsymbol{x}^j|k; \boldsymbol{\psi}_k^j)^{\gamma_2}. \quad (11)$$

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

$$F_3(\Psi|\Gamma) \;=\; \sum_{m=1}^{M} \log \sum_{k=1}^{K} g_k(\boldsymbol{x}_m; \Psi) + \log p(\Psi), \quad (12)$$

where $p(\Psi)$ is a prior over Ψ. We incorporate unlabeled samples into our hybrid classifier directly by maximizing $F_3(\Psi|\Gamma)$ in the same way as for a mixture model in generative approaches, where joint probability models are regarded as discriminative functions.

If Γ is known, we can estimate the Ψ that provides the local maximum of $F_3(\Psi|\Gamma)$ around the initialized value of Ψ, with the help of the EM algorithm [Dempster *et al.*, 1977]. However, Γ is not a known value but an unknown parameter that should be estimated using Eq. (10) with Θ and Ψ fixed. Therefore, we estimate Ψ and Γ iteratively and alternatively. Ψ and Γ are updated until some convergence criterion is met.

4 Experiments

4.1 Test Collections

Empirical evaluations were performed on three test collections: *WebKB* [1], *Cora* [2], and *20newsgroups (20news)* [3] datasets, which have often been used as benchmark tests for classifiers in text classification tasks [Nigam *et al.*, 1999].

WebKB contains web pages from universities. This dataset consists of seven categories, and each page belongs to one of them. Following the setup in [Nigam *et al.*, 1999], we used only four categories: *course, faculty, project*, and *student*. The categories contained 4199 pages. We extracted four components, *Main Text* (MT), *Out-Links* (OL), *In-Links* (IL), and *Anchor-Text* (AT), from each page. Here, MT is the text description except for tags and links. The OL for a page consists of web page URLs linked by the page. The IL for a page is the set of web page URLs linking the page. AT is the anchor text set for each page, which consists of text descriptions expressing the link to the page found on other web pages. We collected IL and AT from the links within the dataset. For the MT and AT components, we removed stop words and vocabulary words included in only one web page. We also removed URLs included in only one page for the OL and IL components. There were 18525 and 496 vocabulary words in the MT and AT of the dataset, respectively. OL and IL contained 4131 and 500 different URLs, respectively.

Cora contains more than 30000 summaries of technical papers, and each paper belongs to one of 70 groups. For our evaluation, we used 4240 papers included in 7 groups: */Artificial_Intelligence/Machine_Learning/**. We extracted three components, *Main Text* (MT), *authors* (AU), and *citations* (CI) from each paper. Here, MT consists of the text distribution included in the papers, and AU is the set of authors. CI consists of citations to other papers. We removed vocabulary words, authors, and cited papers for each component in the same way as for WebKB. There were 9190 vocabulary words, 1495 authors, and 13282 cited papers in the dataset.

20news consists of 20 different UseNet discussion groups. Each article belongs to one of the 20 groups. The test collection has been used to evaluate a supervised hybrid classifier for multi-component data [Raina *et al.*, 2004]. Following the setup in [Nigam *et al.*, 1999], we used only five groups: *comp.**. There were 4881 articles in the 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 content of each article except for the title. We removed stop words and vocabulary words included in only one article. There were 19273 and 1775 vocabulary words in components M and T of the dataset, respectively.

4.2 Experimental Settings

We used a naive Bayes (NB) model [Nigam *et al.*, 2000] as a generative model and a bias correction model for each component, assuming that different features (works, links, cita-

[1] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/webkb-data.gtar.gz

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

[3] http://people.csail.mit.edu/jrennie/20Newsgroups/20news-18828.tar.gz

tions, or authors) included in the component are independent. $\boldsymbol{x}^j$ was provided as the feature-frequency vector of the jth component. For the MAP estimation in Eqs (2), (6), and (12), we used Dirichlet priors as the priors over NB model parameters.

For our experiments, we randomly selected labeled, unlabeled, and test samples from each dataset. We made ten different evaluation sets for each dataset by this random selection. 1000 data samples from each dataset were used as the test samples in each experiment. 2500, 2000, and 2500 unlabeled samples were used with labeled samples for training these classifiers in WebKB, Cora, and 20news, respectively. The average classification accuracy over the ten evaluation sets was used to evaluate the methods.

We compared our H-BCM classifier with three semi-supervised classifiers for multi-component data as mentioned in Section 2: an NB based classifier with EM-λ [Nigam *et al.*, 2000], an NLR classifier with MER (MLR/MER) [Grandvalet and Bengio, 2005], and a CH classifier. We also examined three *supervised* classifiers: NB, MLR, and hybrid (HY) classifiers. NB, MLR, and HY classifiers were trained only on labeled samples.

In our experiments, for EM-λ, the value of weighting parameter λ was set in the manner mentioned in Section 2.2 Note that in our experiments we selected the value from fourteen candidate values of $\{0.01, 0.05, 0.1, 0.2, 0.25, 0.3, 0.4, 0.5, 0.6, 0.7, 0.75, 0.8, 0.9, 1\}$ to save computational time, but we carefully selected these candidate values via preliminary experiments.

For MLR/MER, the value of weighting parameter λ in Eq. (4) was selected from sixteen 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 the preliminary experiments. For a fair comparison of the methods, the value of λ should be determined using training samples, for example, using a cross-validation of labeled samples [Grandvalet and Bengio, 2005]. We determined the value of λ 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.

4.3 Results and Discussion

Table 1 shows the average classification accuracies over the ten different evaluation sets for WebKB, Cora, and 20news for different numbers of labeled samples. Each number in parentheses in the table denotes the standard deviation of the ten evaluation sets. $|D_l|$ ($|D_u|$) represents the number of labeled (unlabeled) samples.

As reported in [Raina *et al.*, 2004; Fujino *et al.*, 2005a], in supervised cases, the hybrid classifier was useful for achieving better generalization performance for multi-component data classification. In our experiments, the average classification accuracy of HY was similar to or better than that of NB and MLR.

We examined EM-λ, MLR/MER, CH, and H-BCM classifiers for semi-supervised cases. The H-BCM classifier outperformed the other semi-supervised classifiers except when the generative classifier performed very differently from the discriminative classifier for WebKB. We confirmed experimentally that the H-BCM classifier was useful for improving the generalization performance of multi-component data classification with both labeled and unlabeled samples.

More specifically, the H-BCM classifier outperformed the MLR/MER classifier except when there were many labeled samples for WebKB. This result is because MLR/MER tends to be overfitted to few labeled samples. In contrast, H-BCM inherently has the characteristic of the generative models, whereby such an overfitting problem is mitigated. When many labeled samples are available such that the overfitting problem can be solved, it would be natural for the discriminative classifier to be better than the H-BCM classifier.

The classification performance with H-BCM was better that with EM-λ except when there were few labeled samples for WebKB. It is known that discriminative approaches often provide better classification performance than generative approaches when there are many labeled samples [Ng and Jordan, 2002]. Our experimental result indicates that there might be an intrinsic limitation preventing EM-λ from achieving a high level of performance because only weighting parameter λ is trained discriminatively.

H-BCM provided better classification performance than CH except when there were few labeled samples for WebKB. The H-BCM and CH classifiers are constructed based on a hybrid of generative and discriminative approaches, but CH differs from H-BCM, in that the former does not contain bias correction models. This experimental result indicates that introducing bias correction models was effective in incorporating unlabeled samples into the hybrid classifier and thus improves its generalization ability.

5 Conclusion

We proposed a semi-supervised classifier design method for multi-component data, based on a hybrid generative and discriminative approach, called Hybrid classifier with Bias Correction Models (H-BCM classifier). The main idea is to design an individual generative model for each component and to introduce models to reduce the bias that results when there are few labeled samples. For evaluation, we also considered straightforward applications of conventional generative and discriminative semi-supervised learning algorithms to the classifiers proposed for multi-component data. We confirmed experimentally that H-BCM was more effective in improving the generalization performance of multi-component data classification than the straightforward approaches. Our experimental results using three test collections suggest that the H-BCM classifier is useful especially when the generative and discriminative classifiers provide similar performance. Future work will involve applying H-BCM to multimodal data in which different generative models are employed.

References

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

[Brochu and Freitas, 2003] E. Brochu and N. Freitas. "name that song!": A probabilistic approach to querying on music and text. In *Advances in Neural Information Processing Systems 15*, pages 1505–1512. MIT Press, Cambridge, MA, 2003.

Table 1: Classification accuracies (%) with classifiers trained on various numbers of labeled samples.

(a) WebKB ($|D_u| = 2500$, $K = 4$)

Training set		Semi-supervised				Supervised		
$\|D_l\|$	$\|D_l\|/\|D_u\|$	H-BCM	CH	EM-λ	MLR/MER	HY	NB	MLR
32	0.0128	69.9 (5.1)	73.2 (3.7)	**73.5** (3.1)	63.5 (5.0)	68.6 (3.7)	68.8 (3.3)	63.3 (4.5)
64	0.0256	76.2 (2.8)	**77.1** (2.5)	76.0 (2.5)	72.2 (2.7)	74.8 (1.1)	73.4 (1.4)	71.9 (2.2)
128	0.0512	**81.8** (1.7)	79.7 (2.2)	77.7 (2.4)	78.5 (2.2)	80.7 (1.9)	77.5 (2.0)	78.3 (2.0)
256	0.1024	**85.0** (1.3)	83.3 (1.1)	80.6 (1.7)	84.3 (1.7)	84.5 (1.3)	80.9 (1.5)	83.9 (1.7)
512	0.2048	87.2 (1.1)	85.0 (1.1)	82.7 (1.2)	**88.7** (1.3)	87.4 (1.1)	83.5 (1.4)	87.9 (1.3)

(b) Cora ($|D_u| = 2000$, $K = 7$)

Training set		Semi-supervised				Supervised		
$\|D_l\|$	$\|D_l\|/\|D_u\|$	H-BCM	CH	EM-λ	MLR/MER	HY	NB	MLR
56	0.028	**76.4** (3.6)	74.1 (5.0)	71.6 (4.2)	55.7 (4.2)	63.2 (3.6)	57.0 (2.9)	55.2 (3.8)
112	0.056	**83.0** (2.1)	80.7 (1.2)	77.9 (1.6)	63.8 (2.3)	72.1 (3.4)	64.5 (3.2)	63.5 (2.1)
224	0.112	**85.7** (1.2)	83.6 (0.9)	81.6 (1.1)	72.9 (1.1)	80.4 (1.1)	75.2 (1.5)	72.3 (1.1)
448	0.224	**87.4** (1.3)	85.6 (1.4)	83.8 (1.0)	78.6 (1.5)	84.4 (1.2)	79.8 (1.2)	77.3 (1.2)
896	0.448	**89.1** (0.7)	88.1 (1.1)	86.8 (1.1)	82.7 (1.2)	87.9 (1.0)	84.6 (0.8)	82.1 (1.2)

(c) 20news ($|D_u| = 2500$, $K = 5$)

Training set		Semi-supervised				Supervised		
$\|D_l\|$	$\|D_l\|/\|D_u\|$	H-BCM	CH	EM-λ	MLR/MER	HY	NB	MLR
40	0.016	**64.8** (3.0)	55.5 (6.2)	53.6 (6.6)	50.2 (5.3)	47.7 (3.7)	45.6 (3.6)	48.8 (3.9)
80	0.032	**71.6** (1.8)	62.5 (4.2)	59.4 (6.4)	57.2 (3.3)	56.3 (2.2)	51.8 (3.6)	56.4 (3.0)
160	0.064	**75.7** (1.5)	70.1 (2.4)	66.3 (4.6)	65.3 (3.2)	65.4 (1.4)	60.1 (3.1)	63.7 (2.3)
320	0.128	**78.5** (1.4)	75.3 (1.8)	71.3 (3.2)	72.3 (1.2)	72.8 (1.8)	67.8 (2.4)	71.0 (1.0)
640	0.256	**82.0** (1.2)	79.4 (1.4)	76.0 (1.5)	77.7 (1.1)	79.6 (1.2)	74.9 (1.6)	76.4 (0.9)
1280	0.512	**85.0** (1.0)	83.4 (1.4)	80.1 (1.8)	81.0 (1.2)	83.4 (0.9)	78.8 (1.5)	80.2 (0.9)

[Chakrabarti *et al.*, 1998] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of ACM International Conference on Management of Data (SIGMOD-98)*, pages 307–318, 1998.

[Chen and Rosenfeld, 1999] S. F. Chen and R. Rosenfeld. A Gaussian prior for smoothing maximum entropy models. Technical report, Carnegie Mellon University, 1999.

[Cohn and Hofmann, 2001] D. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *Advances in Neural Information Processing Systems 13*, pages 430–436. MIT Press, Cambridge, MA, 2001.

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

[Fujino *et al.*, 2005a] A. Fujino, N. Ueda, and K. Saito. A classifier design based on combining multiple components by maximum entropy principle. In *Information Retrieval Technology (AIRS2005 Proceedings), LNCS*, volume 3689, pages 423–438. Springer-Verlag, Berlin, Heidelberg, 2005.

[Fujino *et al.*, 2005b] A. Fujino, N. Ueda, and K. Saito. A hybrid generative/discriminative approach to semi-supervised classifier design. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05)*, pages 764–769, 2005.

[Grandvalet and Bengio, 2005] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems 17*, pages 529–536. MIT Press, Cambridge, MA, 2005.

[Hastie *et al.*, 2001] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, New York Berlin Heidelberg, 2001.

[Joachims, 1999] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning (ICML-99)*, pages 200–209, 1999.

[Lu and Getoor, 2003] Q. Lu and L. Getoor. Link-based text classification. In *IJCAI Workshop on Text-Mining & Link-Analysis (TextLink 2003)*, 2003.

[Ng and Jordan, 2002] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press, Cambridge, MA, 2002.

[Nigam *et al.*, 1999] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

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

[Raina *et al.*, 2004] R. Raina, Y. Shen, A. Y. Ng, and A. McCallum. Classification with hybrid generative/discriminative models. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[Seeger, 2001] M. Seeger. Learning with labeled and unlabeled data. Technical report, University of Edinburgh, 2001.

[Sun *et al.*, 2002] A. Sun, E. P. Lim, and W. K. Ng. Web classification using support vector machine. In *Proceedings of 4th Int. Workshop on Web Information and Data Management (WIDM 2002) held in conj. with CIKM 2002*, pages 96–99, 2002.