Two-view Feature Generation Model for Semi-supervised Learning

Rie Kubota Ando RIE1@US.IBM.COM
IBM T.J. Watson Research Center, Hawthorne, New York, USA

Tong Zhang TZHANG@YAHOO-INC.COM
Yahoo Inc., New York, New York, USA

Abstract

We consider a setting for discriminative semi-supervised learning where unlabeled data are used with a generative model to learn effective feature representations for discriminative training. Within this framework, we revisit the two-view feature generation model of co-training and prove that the optimum predictor can be expressed as a linear combination of a few features constructed from unlabeled data. From this analysis, we derive methods that employ two views but are very different from co-training. Experiments show that our approach is more robust than co-training and EM, under various data generation conditions.

1. Introduction

In many real-world problems, an enormous amount of unlabeled data is available with little effort, while labeled data is costly to obtain. It is thus natural to ask whether, in addition to manually labeled data, one can also take advantage of the unlabeled data. Methods that use both labeled and unlabeled data are generally referred to as *semi-supervised learning*.

We divide earlier efforts into two categories. In the first category, labels for unlabeled data are estimated based on the current classifier maintained by the algorithm. The augmented "labeled" data is then used to retrain the current classifier. Examples of this approach include the transductive SVM (Vapnik, 1998), the co-training method (Blum & Mitchell, 1998), and EM (Nigam et al., 2000, for example). In the second category, the unlabeled data is used to create a good hypothesis space so that a supervised learning algorithm can be applied with the constructed hypothesis space. This approach includes some of the more recent developments such as graph based semi-supervised learning (Zhu et al., 2003, for example) as well as more direct construction of the hypothesis space (Ando & Zhang, 2005, for example).

This paper focuses on the second approach, in which we use unlabeled data to learn good feature representation for discriminative learning. Within this framework, we consider a simple but non-trivial data generation model used earlier for analyzing co-training, where two views of the data are independently generated conditioned on the label.

Our contribution is to show that under this two-view model (but with fewer assumptions than co-training[1]), it is possible to learn a small number of features from unlabeled data so that the optimum predictor can be expressed as a linear combination of these features, even though the original feature space may be high-dimensional. Due to the reduced dimensionality that preserves optimality, the analysis proves the usefulness of unlabeled data for discriminative learning. The result leads to semi-supervised learning methods that employ two views but are very different from co-training. Moreover, it explains the effectiveness of the semi-supervised learning method proposed in (Ando & Zhang, 2005). Experiments show that our approach is more effective than conventional methods under various conditions.

2. Discriminative Semi-supervised Learning and Generative Model

In machine learning, our goal is to predict output $y \in \mathcal{Y}$ given input $x \in \mathcal{X}$, where we assume that $K = |\mathcal{Y}|$ is finite. Assume that (x, y) is drawn from an

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007. Copyright 2007 by the author(s)/owner(s).

[1] That is, we do not assume that each view is sufficient for prediction.

unknown underlying distribution D. In the probability modeling framework, this can be achieved through an estimate of the conditional density $P(y|x)$. Let $P(x, y|\alpha)$ be a family of joint probability distributions on $\mathcal{X} \times \mathcal{Y}$ with unknown parameter α, and assume that it contains D. The supervised learning problem is to estimate the unknown parameter α from labeled examples $\{(X_i, Y_i) : i = 1, \ldots, n\}$ that are independently drawn from D. In semi-supervised learning, we also observe unlabeled data X_j $(j = n+1, \ldots, m)$ that are drawn from D but without the corresponding outputs Y_j.

In this framework, one may decompose $P(x, y|\alpha)$ as $P(x, y|\alpha) = P(y|\alpha, x)P(x|\alpha)$. We shall call the component $P(x|\alpha)$ the *generative component*, and the component $P(y|\alpha, x)$ the *discriminative component*. In many applications, discriminative learning with a model of the form $P(y|\alpha, x)$ is more effective than generative models. This is because x is often high dimensional, so that it is difficult to model $P(x|\alpha)$ well. However, the generative component is useful for learning from unlabeled data. As argued in (Zhang & Oles, 2000), in order for unlabeled data to be useful, it is necessary to incorporate a generative component that depends non-trivially on α.

Following a similar argument, we illustrate this point more generally under the Bayesian decision theoretical framework, where we are given a prior distribution $P(\alpha)$. Under such a prior, the optimal Bayes estimator depends only on the posterior distribution:

$$\begin{aligned} P_{post}(\alpha) = & P(\alpha | \{(X_i, Y_i) : i = 1, \ldots, n\}, \\ & \{(X_j) : j = n+1, \ldots, m\}), \end{aligned}$$

and the Bayes optimal conditional probability is $P(y|x) = \int_\alpha P(y|\alpha, x)\, dP_{post}(\alpha)$. If we redefine a prior on α using unlabeled data as:

$$P_{unlabeled}(\alpha) \propto \prod_{j=1}^{m} P(X_j|\alpha) P(\alpha),$$

then mathematically, we can rewrite the posterior as

$$P_{post}(\alpha) \propto P_{unlabeled}(\alpha) \prod_{i=1}^{n} P(Y_i|\alpha, X_i). \qquad (1)$$

This means that the effect of unlabeled data can be viewed as re-defining a "prior" over α using unlabeled data. With such a redefined prior, we may then apply a discriminative model $P(Y_i|\alpha, X_i)$.

The above derivation is general under the Bayesian framework. Although simple, it shows that from the Bayesian point of view, unlabeled data can only be useful through a redefinition of "prior" (using unlabeled data), combined with a discriminative learning procedure. For example, a practical approach to approximate the posterior of α is to use the δ-function at the MAP (maximum a posterior) estimator:

$$\hat{\alpha} = \arg\min_{\alpha} \left[-\sum_{i=1}^{n} \ln P(Y_i|\alpha, X_i) - \ln P_{unlabeled}(\alpha) \right],$$

and the conditional probability at a data point x is estimated as $P(y|\hat{\alpha}, x)$. Other Bayesian inference procedures can be used as well.

More generally, we may consider a similar framework for discriminative learning in a non-Bayesian setting. Since in a discriminative model, we are interested in finding a function $f_\alpha(x)$ that directly predicts y given x, a prior in the Bayesian setting can be regarded as a restriction on the functional form of the prediction rule $f_\alpha(x)$, or *regularization condition* in the non-Bayesian setting. In this setting, we may drop the parameter α, and consider estimating $f = f_\alpha$ directly in the following regularization method. It is a direct generalization of using MAP to estimate (1):

$$\hat{f} = \arg\min_{f} \left[\sum_{i=1}^{n} \phi(f(X_i), Y_i) + \lambda Q_{unlabeled}(f) \right], \qquad (2)$$

where $\lambda > 0$ is an appropriately chosen regularization parameter, and $\phi(f, y)$ is a loss function which we would like to minimize by fitting f on the training data. The regularization condition $Q_{unlabeled}(f)$ puts restrictions on forms of f in the functional space, and this is the only part in the formulation that depends on unlabeled data.

The regularization formulation (without including unlabeled data) has become standard in modern statistical machine learning. In essence, it replaces the negative log-likelihood loss in the MAP formulation by an arbitrary loss function, and replaces the negative log-prior in MAP by an arbitrary regularization condition which restricts the parametric form of f for the prediction function. Parallel to the Bayesian framework, equation (2) implies that the unlabeled data should be used to construct a regularization condition $Q_{unlabeled}(f)$ for discriminative training. An equivalent view is to construct a hypothesis space $\mathcal{H}_{unlabeled}$ of f using unlabeled data and solve the empirical risk minimization problem: $\hat{f} = \arg\min_{f \in \mathcal{H}_{unlabeled}} \left[\frac{1}{n} \sum_{i=1}^{n} \phi(f(X_i), Y_i) \right]$.

In summary, in order to use unlabeled data in discriminative learning, we can first learn a good regularization condition, or discriminative parameterization form (i.e., hypothesis space) using unlabeled data, and then use the parameterization with a standard discriminative learning method such as MAP, SVM, or boosting etc. The rest of the paper focuses on specific statistical models for which we show how this perspective can be implemented.

3. Two-view Feature Generation Model

Given input space $\mathcal{X}$, we consider two maps (views) $z_1 : \mathcal{X} \to \mathcal{Z}_1$ and $z_2 : \mathcal{X} \to \mathcal{Z}_2$. For simplicity, we also write $z_1(x)$ as z_1 and $z_2(x)$ as z_2. Given (x, y), we assume that the two views $z_1(x)$ and $z_2(x)$ are independent, conditioned on the label y:

$$P(z_1, z_2|y) = P(z_1|y)P(z_2|y). \tag{3}$$

Although it is possible to relax our assumption to weak dependency, we will not analyze it here due to the space limitation. Instead, we will use experiments to show that the performance of the methods developed in this section degrades smoothly when the independence condition becomes violated. This conditional independence assumption is also used in the analysis of co-training in (Blum & Mitchell, 1998; Dasgupta et al., 2001). However, for co-training to be successful, one requires an additional assumption that the label y can be predicted well by z_1 alone and by z_2 alone (view redundancy), which our approach does not assume. Also note that (3) is a weaker assumption than the naive Bayes assumption, which assumes components in x are all independently generated given labels. Such less restrictive assumptions have a practical advantage as shown later in our experiments.

Our goal is to use the generative model based on (3) to obtain (from unlabeled data) features useful for discriminative learning, which implements the idea outlined in Section 2.

3.1. Conditional Formulation

The main trick in our solution is to work with $P(z_1|z_2)$ to obtain conditional estimates of $P(y|z_1)$ and $P(y|z_2)$. As we show later, these estimates can be effectively computed without forming an exhaustive table of $P(z_1|z_2)$ explicitly. The following lemma shows that $P(y|z_1)$ and $P(y|z_2)$ (and $P(y)$) contain all the necessary information for classification.

Lemma 1 *Under the assumption of (3), we have*

$$P(y|z_1, z_2) = c(z_1, z_2)^{-1} P(y|z_1)P(y|z_2)/P(y), \tag{4}$$

where $c(z_1, z_2) = \sum_y P(y|z_1)P(y|z_2)/P(y)$.

Proof Note that

$$P(y|z_1, z_2)P(z_1, z_2) = P(y)P(z_1|y)P(z_2|y) = P(z_1, y)$$
$$P(z_2, y)/P(y) = P(z_1)P(z_2)P(y|z_1)P(y|z_2)/P(y).$$

Since $\sum_y P(y|z_1, z_2) = 1$, we have $c(z_1, z_2) = \sum_y P(y|z_1)P(y|z_2)/P(y)$. ■

The lemma implies that quantities $P(y|z_1)$, $P(y|z_2)$, and $P(y)$ form sufficient statistics for $P(y|z_1, z_2)$.

We can obtain a low-rank decomposition of the conditional probability $P(z_1|z_2)$ from (3):

$$P(z_1|z_2) = \sum_{y \in \mathcal{Y}} P(z_1|y)P(y|z_2).$$

Our goal is to learn $P(y|z_2)$ from this decomposition. Instead of estimating conditional probabilities $P(z_1|z_2)$ on the left-hand side (because it may not be practical to enumerate all possible (z_1, z_2)), we form easier binary-classification problems. Let t_2^ℓ be an arbitrary binary-valued function $\mathcal{Z}_1 \to \{0, 1\}$ for $\ell = 1, \ldots, m$, and for simplicity, denote by $t_2^\ell = t_2^\ell(z_1)$. Then we have:

$$P(t_2^\ell|z_2) = \sum_{y \in \mathcal{Y}} P(t_2^\ell|y)P(y|z_2), \quad \ell = 1, \ldots, m\ . \tag{5}$$

We consider m binary classification problems of predicting $t_2^\ell(z_1)$ (for $\ell = 1, \ldots, m$) from z_2, which we refer to as *auxiliary problems*. By simultaneously solving multiple auxiliary problems, we can obtain a parametric representation of $P(y|z_2)$. Similarly, we can obtain a parametric representation of $P(y|z_1)$. With such representations, we can directly obtain a parametric representation of $P(y|z_1, z_2)$ from (4), which can then be used directly with a discriminative learning algorithm.

In order to obtain the functional form of $P(y|z_1)$ and $P(y|z_2)$ introduced above, we consider an embedding of z_j into a high dimensional Hilbert space $\mathcal{H}_j$ by a feature map $\psi_j : \mathcal{Z}_j \to \mathcal{H}_j$ $(j = 1, 2)$. For notational simplicity[2], we use bold symbol $\mathbf{z}_j$ to represent the vector representation $\psi_j(z_j)$ of z_j. One may also simply assume that ψ_j is the identity operator and $\mathcal{Z}_j = \mathcal{H}_j$.

3.2. Linear Subspace Model

If $\mathcal{H}_2$ is a sufficiently large Hilbert space, then any real-valued function of z_2 can be represented in a form $\vec{\beta}_2^T \mathbf{z}_2$ to arbitrary precision. In particular, $\forall \epsilon > 0$, there exists $\vec{\beta}_2(y) \in \mathcal{H}_2$ such that

$$|P(y|z_2) - \vec{\beta}_2(y)^T \mathbf{z}_2| \leq \epsilon \text{ for all } z_2. \tag{6}$$

[2] Without loss of generality, we may also assume for simplicity that $P(\mathbf{w}^T \mathbf{z}_j = 0) = 1$ implies $\mathbf{w} = 0$ because otherwise, we can simply consider the quotient space $\mathcal{H}_j / \{\mathbf{w} \in \mathcal{H}_j : \mathbf{w}^T \mathbf{z}_j = 0\}$.

The following theorem essentially shows that the optimum predictor can be expressed as a linear combination of at most K features and suggests how to construct this feature space from unlabeled data.

Theorem 1 *Consider the solutions (in the least squares loss sense) of m problems indexed by ℓ in (5): $\mathbf{w}_\ell = \arg\min_{\mathbf{w}} \mathbf{E}_{(z_2,t_2^\ell)}(\mathbf{w}^T\mathbf{z}_2 - t_2^\ell)^2$. Assume (6) with $\epsilon = 0$, and let $B_2 = \mathrm{span}(\{\vec{\beta}_2(y) : y \in \mathcal{Y}\})$, where $\vec{\beta}_2(y)$ satisfies $P(y|z_2) = \vec{\beta}_2(y)^T\mathbf{z}_2$ for any z_2. Then $\mathbf{w}_\ell \in B_2$, thus the rank of $\mathrm{span}(\{\mathbf{w}_\ell\})$ is at most K.*

In the non-degenerate situation where the rank is K, let $\{\mathbf{v}_k\}$ $(k = 1, \ldots, K)$ be a set of basis vectors of $\mathrm{span}(\{\mathbf{w}_\ell\})$. Then there exist $\gamma_k(y)$ (for $k = 1, \ldots, K$) such that $P(y|z_2) = \sum_{k=1}^K \gamma_k(y)\mathbf{v}_k^T\mathbf{z}_2$ for any y and z_2.

Proof We have $P(y|z_2) = \vec{\beta}_2(y)^T\mathbf{z}_2$. Let $P(t_2^\ell = 1|y) = \alpha_2^\ell(y)$, we obtain from (5): $P(t_2^\ell = 1|z_2) = \sum_{y\in\mathcal{Y}} \alpha_2^\ell(y)\vec{\beta}_2(y)^T\mathbf{z}_2$. That is,

$$\sum_{y\in\mathcal{Y}} \alpha_2^\ell(y)\vec{\beta}_2(y) = \arg\min_{\mathbf{w}} \mathbf{E}_{(z_2,t_2^\ell)}(\mathbf{w}^T\mathbf{z}_2 - t_2^\ell)^2.$$

Therefore we have $\mathbf{w}_\ell = \sum_{y\in\mathcal{Y}} \alpha_2^\ell(y)\vec{\beta}_2(y) \in B_2$. This proves the first part of the theorem. For the second part, since B_2 has rank at most K, if the set of $\mathbf{w}_\ell$ has rank K, then $B_2 = \mathrm{span}(\{\mathbf{w}_\ell : \ell = 1, \ldots, m\}) = \mathrm{span}(\{\mathbf{v}_k : k = 1, \ldots, K\})$. It follows that $\vec{\beta}_2(y)$ can be represented as $\sum_k \gamma_k(y)\mathbf{v}_k$. This implies that $P(y|z_2) = \sum_{k=1}^K \gamma_k(y)\mathbf{v}_k^T\mathbf{z}_2$. ■

The theorem essentially says that the K feature functions $\mathbf{v}_k^T\mathbf{z}_2$ $(k = 1, \ldots, K)$ give sufficient statistics for $P(y|z_2)$. Therefore basis vectors of $\mathrm{span}(\{\mathbf{w}_\ell\})$, obtained by SVD of the matrix of vectors $\mathbf{w}_\ell$, can be used to produce good K-dimensional feature vectors containing all the information needed for classification.

Above, we only proved the theorem under exact assumptions (conditional independence and $\epsilon = 0$). Though the page limilation precludes detail, it can be shown, using perturbation analysis, that when the assumptions are moderately violated, we can still obtain useful basis vectors (which approximately span B_2) using SVD and keeping the most significant dimensions.

This SVD-based feature generation method suggested by our analysis is, in fact, essentially the same as the scheme proposed in (Ando & Zhang, 2005). Thus, Theorem 1 explains why (Ando & Zhang, 2005)'s semi-supervised learning method is effective ((Ando & Zhang, 2005) did not analyze why using unlabeled data in the way proposed there can be helpful). Theorem 1 proves the effectiveness of semi-supervised learning under the statistical model of (3), and provides a concrete example for the idea outlined in Section 2.

Exchanging the roles of the two views and using $\mathbf{w}'_\ell = \arg\min_{\mathbf{w}} \mathbf{E}_{(z_1,t_1^\ell)}(\mathbf{w}^T\mathbf{z}_1 - t_1^\ell)^2$, we can compute the basis functions $\mathbf{u}_k$ $(k = 1, \ldots, K)$ that spans $\mathbf{w}'_\ell$. Under assumptions analogous to Theorem 1, we obtain $P(y|z_1) = \sum_{k=1}^K \gamma'_k(y)\mathbf{u}_k^T\mathbf{z}_1$. Now, from Lemma 1, $P(y|z_1)$ and $P(y|z_2)$ give sufficient statistics for the classification problem. Therefore $P(y|z_1, z_2)$ can be expressed using features $\mathbf{u}_k^T\mathbf{z}_1$ and $\mathbf{v}_k^T\mathbf{z}_2$. One may directly use a linear combination of feature vectors $[\mathbf{u}_k^T\mathbf{z}_1]$ and $[\mathbf{v}_k^T\mathbf{z}_2]$, which gives a feature vector of size $2K$. If y can be predicted well from either $P(y|z_1)$ alone or $P(y|z_2)$ alone, then it can be predicted well from a linear classifier based on these $2K$ features. In the case that a nonlinear combination of $P(y|z_1)$ and $P(y|z_2)$ is more effective, we obtain from (4) that $P(y|z_1, z_2) = \sum_{k,k'=1}^K \beta_{k,k'}(y)\mathbf{u}_k^T\mathbf{z}_1\mathbf{v}_{k'}^T\mathbf{z}_2$, where $\beta_{k,k'}(y) = \gamma_{k'}(y)\gamma'_k(y)/P(y)$.

The parameters $\beta_{k,k'}$ can be obtained by discriminative learning on labeled data such as least squares regression:

$\arg\min_{\vec{\beta}} \sum_{i=1}^n \sum_{y'\in\mathcal{Y}} \left(\vec{\beta}(y')^T\mathbf{z}(X_i) - I(y' = Y_i)\right)^2$, where $\mathbf{z}(x) = [\mathbf{u}_k^T\mathbf{z}_1\mathbf{v}_{k'}^T\mathbf{z}_2]_{k,k'=1,\ldots,m} \in R^{K^2}$. The formulation has K^3 parameters $\beta_{k,k'}(y)$ for $k, k' = 1, \ldots, K$ and $y \in \mathcal{Y}$. The disadvantage of this method, which uses products $\mathbf{v}_k^T\mathbf{z}_1\mathbf{v}_{k'}^T\mathbf{z}_2$ instead of $\mathbf{v}_k^T\mathbf{z}_1$ and $\mathbf{v}_{k'}^T\mathbf{z}_2$ separately, is that the dimensionality becomes K^3 instead of $2K^2$. Because of the increased dimensionality, it may not perform as well as using $\mathbf{v}_k^T\mathbf{z}_1$ and $\mathbf{v}_{k'}^T\mathbf{z}_2$ as separate features in a linear classifier, when each view is sufficient by itself. However, it still has theoretical significance because it provides a K^3 parameter representation of the target function that is impossible to obtain without unlabeled data.

3.3. Log-linear Mixture Model

In place of (6) where the conditional probability is approximated as a linear combination of features, we may assume that the conditional class probability is a log-linear combination of features:

$$P(y|z_2) \propto \exp(\vec{\beta}_2(y)^T\mathbf{z}_2). \qquad (7)$$

This leads to a more probabilistic procedure to maximize the likelihood of the model. Let $P(t_2^\ell = 1|y) = \alpha_2^k(y)$, we can obtain from (5) the following equation

$$P(t_2^\ell = 1|z_2) = \frac{\sum_{y\in\mathcal{Y}} \alpha_2^\ell(y)\exp(\vec{\beta}_2(y)^T\mathbf{z}_2)}{\sum_{y\in\mathcal{Y}} \exp(\vec{\beta}_2(y)^T\mathbf{z}_2)}$$

for each of the m problems. Instead of treating each binary auxiliary problem separately, we may also consider an m-valued function $t_2 : \mathcal{Z}_1 \rightarrow \{1, \ldots, m\}$ (instead of m binary-value functions t_2^ℓ) and let $P(t_2(z_1) = \ell | y) = \alpha_2^\ell(y)$. Then

$$P(t_2(z_1) = \ell | z_2) \quad = \quad \frac{\sum_{y \in \mathcal{Y}} \alpha_2^\ell(y) \exp(\vec{\beta}_2(y)^T \mathbf{z}_2)}{\sum_{y \in \mathcal{Y}} \exp(\vec{\beta}_2(y)^T \mathbf{z}_2)} \, .$$

For this model, we can use unlabeled data to solve

$$\max_{\alpha_2, \mathbf{w}} \mathbf{E}_{(z_1, z_2)} \left[\ln \frac{\sum_{y \in \mathcal{Y}} \alpha_2^{t_2(z_1)}(y) e^{\mathbf{w}(y)^T \mathbf{z}_2}}{\sum_{y \in \mathcal{Y}} e^{\mathbf{w}(y)^T \mathbf{z}_2}} \right], \tag{8}$$

subject to the condition $\sum_{\ell=1}^m \alpha_2^\ell(y) = 1$ for each $y \in \mathcal{Y}$. This optimization can be solved by EM.

Once we obtain the K vectors $\mathbf{w}(y)$ for $y \in \mathcal{Y}$ and let $\mathbf{v}_k = \mathbf{w}(y_k)$ $(k = 1, \ldots, K)$, we know that in the non-degenerate case, $\mathbf{v}_k = \vec{\beta}_2(y)$ up to a permutation (note that we are unable to find the correspondence of k to y just from (8)). A theorem similar to Theorem 1 can be obtained, which we shall skip due to the space limitation. Similarly, by assuming $P(y|z_1) \propto \exp(\vec{\beta}_1(y)^T \mathbf{z}_1)$, we can obtain the solution $\vec{\beta}_1(y)$ for $y \in \mathcal{Y}$ and let $\mathbf{u}_k$ $(k = 1, \ldots, K)$ correspond to a permutation of $\vec{\beta}_1(y)$ $(y \in \mathcal{Y})$. We can then obtain the following parametric form:

$$\begin{aligned} & P(y|z_1, z_2) \propto P(y|z_1) P(y|z_2) / P(y) \\ & \propto \exp \left[\gamma_0(y) + \sum_{k=1}^K (\gamma_k(y) \mathbf{u}_k^T \mathbf{z}_1 + \gamma'_k(y) \mathbf{v}_k^T \mathbf{z}_2) \right], \end{aligned}$$

which contains $2K^2 + K$ parameters.

In this model, after learning $\{\mathbf{u}_k\}$ and $\{\mathbf{v}_k\}$, we have a simple linear discriminative parameterization of the problem with $2K^2 + K$ features. By Lemma 1, we can use maximum entropy to solve γ using labeled data:

$$\begin{aligned} P(y|z_1, z_2) & \propto \exp(\gamma(y)^T \mathbf{z}(x)); \\ \gamma & = \arg\max_\gamma \sum_{i=1}^n \ln \frac{\exp(\gamma(Y_i)^T \mathbf{z}(X_i))}{\sum_{y' \in \mathcal{Y}} \exp(\gamma(y')^T \mathbf{z}(X_i))}, \\ \mathbf{z}(x) & = [1, \mathbf{u}_1^T \mathbf{z}_1, \ldots, \mathbf{u}_K^T \mathbf{z}_1, \mathbf{v}_1^T \mathbf{z}_2, \ldots, \mathbf{v}_K^T \mathbf{z}_2]. \end{aligned}$$

Compared to the linear subspace model, the advantage of this method is that only $2K^2 + K$ parameters (instead of K^3) are required to express the target $P(y|z_1, z_2)$ linearly. The potential disadvantage is that the EM for solving (8) may get stuck in a local minimum.

4. Experiments

We test the semi-supervised learning methods derived from the two models discussed above in comparison with alternatives: co-training and EM.

4.1. Methods

Linear subspace model (LS) The implementation of the linear subspace model-based method (hereafter, *LS*) follows Section 3.2. Suppose that we are given two views $\mathcal{Z}_i$ and m binary-valued functions t_i^ℓ for $i \in \{1, 2\}$, a labeled data set L, and a unlabeled data set U. First, obtain m weight vectors by solving $\mathbf{w}_\ell = \arg\min_\mathbf{w} \sum_{(z_1, z_2) \in U} (\mathbf{w}^T \mathbf{z}_2 - t_2^\ell(z_1))^2$ for $\ell = 1, \ldots, m$. Second, let $\mathbf{W}$ be a matrix whose columns are the m weight vectors, and compute $\mathbf{v}_2^1, \cdots, \mathbf{v}_2^p$ to be $\mathbf{W}$'s most significant left singular vectors, where p is a dimensionality parameter. Third, exchange the roles of two views and compute $\mathbf{v}_1^1, \ldots, \mathbf{v}_1^p$. Fourth, given $(\mathbf{z}_1, \mathbf{z}_2)$, generate a $2p$-dimensional feature vector whose components are $\mathbf{z}_i^T \mathbf{v}_i^j$ for $i = 1, 2$ and $j = 1, \ldots, p$. Our new feature vector is a concatenation of $\mathbf{z}_1, \mathbf{z}_2$, and the $2p$-dimensional vector computed. Train the final classifier with labeled data L using the new feature vector representation.

As we have discussed, instead of the $2p$-dimensional vector, one may produce a p^2-dimensional vector containing $(\mathbf{z}_1^T \mathbf{v}_1^i)(\mathbf{z}_2^T \mathbf{v}_2^j)$ for $i = 1, \ldots, p$ and $j = 1, \ldots, p$, which represents the interaction of two views. In idealized cases, the dimensionality p should be set to the number of classes K. In practice, p should be determined by cross validation on labeled data as the number of inherent sub-classes underlying the data may not be K.

Log-linear mixture model (LLM) Our experiments use the m-valued function-based formulation and optimize (8). This optimization is done by the standard EM procedure, where we introduce a hidden indicator variable $\xi_{z_1, z_2}(y)$ that is one if the data point (z_1, z_2) has label y; zero otherwise. In the E-step, we compute the expectation of this variable for each data point and for each $y \in \mathcal{Y}$ given α_2 and $\mathbf{w}$. In the M-step, we update $\mathbf{w}$ by solving standard maximum entropy and update α_2, given probabilistic (soft) labels weighted proportionally to the expectation of $\xi_{z_1, z_2}(y)$. We initialize the EM procedure by setting $\alpha_2^\ell(y) = 1/m$ (uniform) and initializing weight vectors $\mathbf{w}(y)$ using labeled data.

Auxiliary problems (function t_2) In our experiments, we define function $t_2(z_1)$ to indicate which entry of $\mathbf{z}_1$ (vector representation of z_1) has the largest value while breaking ties by preferring a smaller vector entry index.

Co-training (baseline) Our implementation follows the original work (Blum & Mitchell, 1998). The two classifiers (employing distinct feature maps) are trained with labeled data. We maintain a pool of q unlabeled instances by random selection. The classifier proposes labels for the instances in this pool. We choose s instances for each classifier with high confidence while preserving the class distribution observed in the initial labeled data, and add them to the labeled data. The process is then repeated, and the final classifier is trained using both views. We use a discriminative linear classifier, which is also used for LS and the supervised baseline.

Naive Bayes EM (baseline) Our implementation of the naive Bayes-based EM essentially follows (Nigam et al., 2000). At a high level, this EM process can be summarized as follows. The model parameters (estimates of $P(y)$ and $P(f|y)$ where f is a feature component) are initialized by the labeled data, and probabilistic labels (soft labels) are assigned to the unlabeled data points according to the model parameters. Using the assigned soft labels, the model parameters are updated based on the naive Bayes assumption, and the process repeats.

4.2. Synthesized Data Experiments

This section reports experiments using artificial data sets generated by controlling properties of the two views. This shows the behavior of different algorithms when different data assumptions are violated.

4.2.1. Data Generation

We generate data sets as follows. First we generate two sets of tokens (F_1 and F_2) of size s (vocabulary) corresponding to two views, respectively. For each class $y \in \{1, \dots, K\}$, two token arrays $T_{1,y}[0..t-1]$ and $T_{2,y}[0..t-1]$ of size t (class vocabulary) are generated by randomly drawing (without replacement) t times from F_1 and F_2, respectively. To generate a data point, we pick a class y randomly from $\{1, \dots, K\}$, and then draw tokens q_1 times from $T_{1,y}$ and draw tokens q_2 times from $T_{2,y}$. Feature vectors are generated based on the 'bag of words'.

We create data sets of various properties. To generate a data set that satisfies the naive Bayes assumption, we draw tokens from $T_{1,y}$ and $T_{2,y}$ randomly and independently. To introduce dependency among the q_i tokens belonging to view-i, we generate a number $p_i \in [0, t-1]$ randomly and then choose q_i consecutive tokens in the array $T_{i,y}$ (i.e. $T_{i,y}[p_i], \dots, T_{i,y}[(p_i + q_i - 1) \mod t]$). In order to additionally introduce dependency between the two views, we set $p_1 = p_2$ with probability r, where r is a parameter to control the degree of the dependency between the two views.

We fix the number of classes K to 10 and generate 30K data points for each data set. Each data set is randomly divided into three sets: the training set of 100 labeled consisting of data points, the test set of 1000 data points, and the remaining 29000 data points as the unlabeled set. We fix the dimensionality p for LS to 10, the number of classes, unless otherwise specified. We report the average and standard deviation (errorbar) of five runs generated with the same configuration but different random seeds.

4.2.2. Results

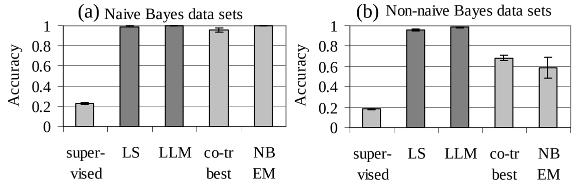

Figure 1. (a) On naive Bayes data sets. (b) On non-naive Bayes data sets.

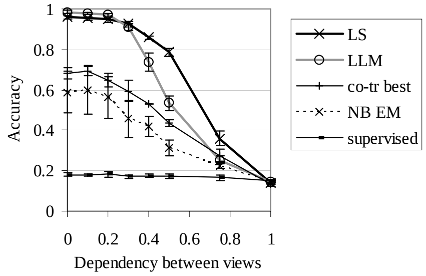

Figure 2. x-axis: r – probability that features from two views have dependency on each other.

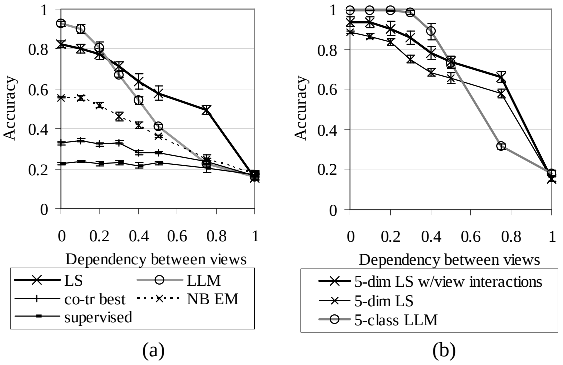

Figure 3. With non-redundant views: (a) Standard comparisons. (b) Dimensionality; interaction features.

Naive Bayes data: Figure 1 (a) shows classification accuracy on the data sets that satisfy the naive Bayes assumption on the feature components. These data sets satisfy the model assumptions of all the tested semi-supervised methods. Consequently, they all perform extremely well, producing nearly 100% accuracy while the supervised baseline (a discriminative linear classifier) achieves only 23%.

Non-naive Bayes data: The naive Bayes assumption is often unrealistic. The data sets used in Figure 1 (b) (and all others) were generated in a more realistic manner so that features belonging to the same view are dependent on each other. In this experiment, the two views are still kept conditionally independent given classes. Our LS and LLM perform well. The performance of naive Bayes EM degrades, reflecting the violation of its model assumption.

Dependency between two views: Figure 2 investigates the effect of violating the conditional independence assumption of the two views. The degree of violation is controlled by parameter r (probability that an instance is generated so that the features from the two views are dependent on each other; see Section 4.2.1). As the dependency between the two views increase (i.e., as r increases), the performance of all the semi-supervised methods degrade. However, even when $r = 0.3$, LS and LLM still achieve $> 90\%$ accuracy. When two views are completely dependent ($r = 1$), all methods fail to benefit from unlabeled data.

Non-redundant views: It is known that for co-training to work well, each of the two views should be sufficient for classification by itself, which is sometimes referred to as the *view redundancy assumption.* The data sets used in Figure 3 violate this assumption in that to discriminate the 10 classes, both views have to be used. That is, view 1 discriminates 5 super classes (class1&2, ..., 9&10), and view 2 discriminates 5 super classes of other combinations (class1&6, ..., 5&10). (Therefore, using one view alone would achieve 50% accuracy at best.) Data generation was done by letting the classes that compose one super class share the same class vocabulary array (setting $T_{1,y_1} = T_{1,y_2}, T_{2,y_1} = T_{2,y_6}$, and so forth). Note that non-redundant views are common in NLP tasks such as named entity chunking. As shown in Figure 3 (a), co-training does not perform well. It underperforms EM while it was outperforming EM when the redundancy assumption was met (Figure 2). By contrast, LS and LLM perform well as long as the dependency between views is low. This is consistent with our theoretical analysis.

Experiments so far have fixed the dimensionality for LS to 10 (the number of classes). However, since each view was generated from 5 super classes on these data sets, one would expect that dimensionality 5 would be more suitable. Moreover, the interactions of two views are expected to be useful on non-redundant views because in the LS model, the optimum predictor cannot be expressed as a simple linear combination of features from the two views separately. This is confirmed in Figure 3 (b); observe that '5-dim LS w/view interactions' outperforms '5-dim LS' in (b), which outperforms 'LS' (10-dim) in (a). Furthermore, '5-class LLM' in (b) performs well, for which EM was initialized using the data labeled with 5 super classes.

LS versus LLM: Throughout the synthesized data experiments, we have observed that LLM generally outperforms LS when the model assumptions are close to being satisfied, for example, when the dependency between two views is low ($r < 0.3$). LLM's superiority is most prominent on the non-redundant views (Figure 3). This is consistent with our analysis; that is, the fact that LLM requires fewer parameters than LS ($2K^2 + K$ vs. K^3) to express non-linear view interactions is critical in this setting. However, when the view dependency becomes higher, LLM's performance drops more rapidly than LS, showing more sensitivity to the violation of the model assumptions. This is possibly because the optimization process is tightly coupled with the probabilistic model assumptions, unlike LS which employs SVD.

4.3. Experiments on Real-world Data

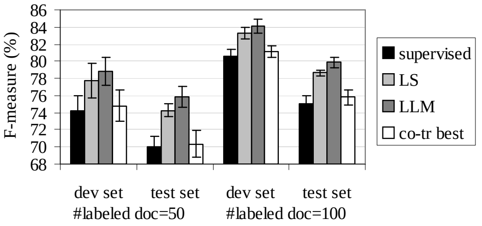

Figure 4. Named entity chunking performance.

Due to the space limitation, we only include one real example. Note that in (Ando & Zhang, 2005), the effectiveness of a method similar to LS has already been shown on a number of tasks including text categorization, part-of-speech tagging, and hand-written digit image classification.

4.3.1. Named Entity Chunking

We use named entity chunking data set (English) provided for the CoNLL'03 shared-task[3]. The corpus is annotated with four types of named entities: persons, organizations, locations, and miscellaneous names. We use the official training/development/test splits and 2

[3] http://cnts.uia.ac.be/conll2003/ner

million words of Reuters articles (unlabeled data). The chunking problem is cast as sequential labeling by encoding chunk information into word tags. Our feature representation and the decoding algorithm follow those of (Ando & Zhang, 2005), and the detail isn't important for the purpose of the paper. We learn new features from unlabeled data using four types of two-view configurations derived from 'current word' vs. 'previous word' and 'current word' vs. 'next word'. That is, in the first type, the auxiliary problems are to predict the current word based on the previous word; in the second type, the auxiliary problems are to predict the previous word based on the current word; and so forth.

Because the naive Bayes classifier produced significantly lower performance than our supervised baseline either in the supervised or semi-supervised setting, we do not include the results.

4.3.2. Results

We use 50 (and 100) documents as training data randomly drawn from the official training set (consisting of 945 documents) and evaluate performance in F-measure of name chunk detection on the development set and the test set. We conduct 5 runs and report the average performance in Figure 4. Both LS and LLM significantly improve performance over the supervised baseline. Co-training does not perform well on this task even though we give unfair advantage to it by optimizing parameters including the number of iterations. For co-training, the feature split 'current+left-context' vs. 'current+right-context' was used, which was better than 'current' vs. 'left context' (or 'right context').

Natural feature splits on this task appear to provide the views whose degree of conditional independence is high enough for our LS and LLM to perform well. However, each view is, apparently, not sufficient for classification by itself[4], and so co-training fails when the split like 'current' vs. 'left context' is used. If we use overlapping views such as 'current+left-context' vs. 'current+right-context', each view becomes more informative, but also dependency between views increases, which degrades performance. Thus, our approach has a clear advantage over co-training on this type of task.

[4] For example, knowing the next word is "said" is not sufficient for deciding whether the current word is a person name or an organization name.

5. Conclusion

We presented a framework for semi-supervised learning, where a generative model is used to learn effective parametric feature representations for discriminative learning. Using the two-view model as in co-training but without assuming view redundancy, we proved (under the ideal model assumptions) that one can construct a small set of features from unlabeled data so that the optimum predictor can be represented as a linear combination of these features. The result immediately implies the effectiveness of this approach to semi-supervised learning.

Our experiments demonstrated that when the conditional independence assumption is violated, the performance of our methods (derived from our analysis) degrades smoothly. Since the assumptions in our model are less restrictive than those of co-training and naive Bayes EM, our approach can perform well even when co-training and naive Bayes fail. We validated the claim empirically by investigating effect of the violation of different data generation assumptions. Finally, our theoretical analysis leads to a satisfactory explanation of the effectiveness of a related method proposed in (Ando & Zhang, 2005).

References

Ando, R. K., & Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, *6*, 1817–1853.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *Proceedings of the eleventh annual conference on Computational learning theory* (pp. 92–100).

Dasgupta, S., Littman, M., & McAllester, D. (2001). PAC generalization bounds for co-training. *NIPS' 01*.

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

Vapnik, V. (1998). *Statistical learning theory*. New York: John Wiley & Sons.

Zhang, T., & Oles, F. J. (2000). A probability analysis on the value of unlabeled data for classification problems. *ICML 2000* (pp. 1191–1198).

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. *ICML 2003*.