Semi-Supervised Classification with Hybrid Generative/Discriminative Methods

Gregory Druck, Chris Pal
Dept. of Computer Science
U. of Massachusetts, Amherst
{gdruck,pal}@cs.umass.edu

Xiaojin Zhu
Computer Sciences Dept.
U. of Wisconsin-Madison
jerryzhu@cs.wisc.edu

Andrew McCallum
Dept. of Computer Science
U. of Massachusetts, Amherst
mccallum@cs.umass.edu

ABSTRACT

We compare two recently proposed frameworks for combining generative and discriminative probabilistic classifiers and apply them to semi-supervised classification. In both cases we explore the tradeoff between maximizing a discriminative likelihood of labeled data and a generative likelihood of labeled and unlabeled data. While prominent semi-supervised learning methods assume low density regions between classes or are subject to generative modeling assumptions, we conjecture that hybrid generative/discriminative methods allow semi-supervised learning in the presence of strongly overlapping classes and reduce the risk of modeling structure in the unlabeled data that is irrelevant for the specific classification task of interest. We apply both hybrid approaches within naively structured Markov random field models and provide a thorough empirical comparison with two well-known semi-supervised learning methods on six text classification tasks. A semi-supervised hybrid generative/discriminative method provides the best accuracy in 75% of the experiments, and the *multi-conditional learning* hybrid approach achieves the highest overall mean accuracy across all tasks.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Experimentation, Performance

Keywords

Semi-supervised learning, hybrid generative/discriminative methods, text classification

KDD'07, August 12–15, 2007, San Jose, California, USA.

1. INTRODUCTION

Most machine learning methods rely on the availability of large labeled datasets. However, human annotation is time-consuming, making labeled data costly to obtain in practice. Motivated by this problem, researchers have proposed *semi-supervised* learning methods that leverage large amounts of relatively inexpensive unlabeled data along with small amounts of labeled data. The increasing interest in applying machine learning to new domains and the vast availability of unlabled data from the web and elsewhere are driving interest in semi-supervised learning.

Semi-supervised learning is especially relevant for applications in data mining, as when initially analyzing data from a new domain, obtaining any labeled data requires laborious human annotation. The lowest effort approach to data mining would use unsupervised learning. However, supervised learning methods typically provide better, more task-focused results.

For example, consider the problem of classifying messages as belonging to a *mac* hardware or *pc* hardware newsgroup. Although there are word features in the data relevant to this task (such as "powerbook" indicating *mac*, or "dell" indicating *pc*), because *mac* and *pc* postings have high word overlap, an unsupervised clustering algorithm could discover many different ways to partition this data. For example, messages about hard drives or networking may appear as clusters, but these clusters may not be directly relevant to the classification task of interest. If posed as a supervised classification task, however, with labeled examples from each newsgroup, the classifier will learn to focus on the features relevant to the $mac - pc$ task, and make the desired separation.

Training methods for machine learning classifiers are often characterized as being generative or discriminative. Generative training estimates the joint distribution of all variables, both the classes and the "input" data, while discriminative training is concerned only with the decision boundary. Classifiers trained discriminatively seem to have lower asymptotic error than analogous generatively-trained classifiers because, intuitively, they are able to focus limited representational capacity on predicting just the class variable. However, discriminative approaches do not always provide the highest accuracy. When the amount of training data is small, generative training can provide better accuracy even when the model is not a very good fit to the data. Ng and Jordan demonstrate this, comparing naive Bayes and logistic regression [15].

Motivated by these observations, several researchers have proposed hybrid methods that combine generative and discriminative training. These hybrid methods have delivered promising results in the domains of text classification [13, 18], pixel classification [10], and object recognition [21, 11], among others.

A variety of semi-supervised techniques have been developed for both generative and discriminative models. A straightforward, generative semi-supervised method is the expectation maximization (EM) algorithm. The EM approach for naive Bayes text classification models is discussed by Nigam et al. in [17]. Generative semi-supervised methods rely on a model for the distribution of the input data, and can fail either when this model is wrong, or when the structure of the input data is not correlated with the classification task (as illustrated in the *mac-pc* example above). Discriminative semi-supervised methods, including probabilistic and non-probabilistic approaches, such as transductive or semi-supervised support vector machines (TSVMs, S3VMs) [8, 7] and a variety of other graph based methods [22, 1] assume high density within class and low density between classes, and can fail when the classes are strongly overlapping. Hence, these approaches for semi-supervised learning in discriminative classifiers also use model assumptions about the structure of the input data, but typically do not encode these assumptions as explicit models of input probability density.

In this paper, we apply hybrid generative/discriminative methods to semi-supervised learning. We compare two recently proposed approaches to combining generative and discriminative methods in detail. The first is multi-conditional learning [13], a class of training objective functions composed of the product of multiple likelihoods that share one set of parameters and are derived from an underlying joint model. We formulate the semi-supervised training problem in terms of the optimization of a multi-conditional objective function that is a weighted combination of a discriminative likelihood of labeled data and a marginal likelihood of both labeled and unlabeled data. We also consider a framework proposed by Lasserre et al. [11] which we henceforth refer to as the *parameter coupling prior*[1] method. In this approach, the discriminative and generative components derived from a common joint model have separate sets of parameters. These parameters are coupled by a prior distribution that specifies how one set of parameters influences the other.

Both of these hybrid approaches can be interpreted as discriminative classifiers trained using the marginal likelihood of the input data as parameter regularization. We conjecture that for many problems this form of regularization is more helpful than typical discriminative regularization approaches penalizing decision boundaries passing through regions of high marginal density. In contrast, these generative/discriminative hybrids are not constrained to avoid low-density regions between classes when placing decision boundaries. Additionally, they are able to balance between leveraging innate clusters in the input data (which may or may not be useful) and task-specific evidence from the labeled data (which may or may not be representative). Hybrid methods can avoid relying on generative modeling assumptions by emphasizing the discriminative likelihood during maximization. In summary, these methods allow us to avoid the assumptions of discriminative semi-supervised approaches and mitigate the assumptions of generative semi-supervised methods. By emphasizing each component of the objective function appropriately, they allow semi-supervised learning in cases that other methods fail.

In addition to the motivation provided above, the contributions of this paper are:

- We apply multi-conditional learning to semi-supervised learning.
- We compare the multi-conditional learning approach with a framework recently proposed by Lasserre, et al. in [11]. We subject this method to much more thorough evaluation than was provided in [11] (only one dataset, no comparisons to other methods).
- We implement these model-independent approaches in a naively structured Markov random field model and derive the appropriate gradients.
- We provide an empirical comparison of the two approaches along with two other prominent semi-supervised methods for classification. A hybrid method outperforms other methods in 75% of the experiments and the multi-conditional learning approach gives the highest overall mean accuracy.

2. TWO GENERAL GENERATIVE-DISCRIMINATIVE APROACHES

First, we define the learning problem. Suppose we have data $D = D_L \cup D_U$, where D_L and D_U represent the labeled and unlabeled data, respectively. Each example in D_L is a pair $(\mathbf{y}, \mathbf{x})$, where the vector $\mathbf{y}$ has length equal to the number of classes and a 1 in the position corresponding to the index of the correct class (other entries are 0). Vector $\mathbf{x}$ has length equal to the number of features of the input and each position contains the value of a particular feature for this example. In D_U, each example is only $(\mathbf{x})$, as the value of $\mathbf{y}$ is hidden. In the case of document classification, for example, each example corresponds to a document, and each position in $\mathbf{x}$ might contain the number of times a particular word occurs.

Notice that $\mathbf{x}$ can be decomposed into $\sum_i^N \mathbf{w}_i$, where $N = \sum_i^{|\mathbf{x}|} \mathbf{x}_i$, so that each $\mathbf{w}_i$ corresponds to the event of observing a feature. Vector $\mathbf{w}_i$ has a single 1 in one position and 0 elsewhere. For example, in document classification $\mathbf{w}_i$ represents a word occurrence. Another occurrence of the same word in the document would correspond to separate event $\mathbf{w}_k$. This decomposition of $\mathbf{x}$ into individual events is useful for understanding the graphical model introduced in Section 3. First, we discuss two model-independent hybrid approaches.

2.1 Multi-Conditonal Learning

Multi-conditional learning [13] is a class of training objective functions composed of the product of multiple weighted likelihoods, each with parameters derived from the same underlying joint model. An advantage of the multi-conditional

[1]We note that *parameter coupling prior* is a short name we devised to refer to the work of Lasserre, Bishop, and Minka. This term is not used in their paper.

framework is the flexibility it allows to craft an objective function for a specific learning task. For example, an objective function composed of the product of weighted discriminative likelihoods for multiple tasks is a natural framework for transfer learning or multitask learning [5].

McCallum et al. [13] combine discriminative and generative likelihoods using the multi-conditional objective function:

$$P(Y|X;\Theta)^{\alpha}P(X|Y;\Theta)^{\beta}$$

Training text classification models with this objective function was found to produce improvements in classification accuracy. Here, we express semi-supervised training in terms of a multi-conditional objective function by combining the weighted discriminative likelihood of the labeled data and the weighted marginal likelihood of labeled and unlabeled data. This objective function is:

$$\mathcal{O}(\Theta) = P(Y_L|X_L;\Theta)^{\alpha}P(X;\Theta)^{\beta},$$

where X_L and Y_L denote the labeled data and the term $P(X;\Theta)$ includes both labeled and unlabeled data.

It is convenient to maximize the natural log of $\mathcal{O}$:

$$\ln \mathcal{O}(\Theta) = \alpha \ln P(Y_L|X_L;\Theta) + \beta \ln P(X;\Theta) \quad (1)$$

We choose the model parameters $\hat{\Theta}$ that maximize $\mathcal{O}$:

$$\hat{\Theta} = \arg\max_{\Theta}(\ln \mathcal{O}(\Theta)).$$

In equation (1), increasing α gives more weight to the discriminative component during maximization, while increasing β gives more weight to the generative component.

A practical concern is that each component and its gradient may be different in scale. Notice that $P(Y_L|X_L;\Theta)$ is a distribution is over the number of labels, and $P(X;\Theta)$ is a distribution over the number of features. This means that if the distributions were uniform the magnitude of the log-likelihood for the generative component would be much smaller than that of the discriminative component. Additionally, in semi-supervised learning the number of labeled examples is typically much smaller than the number of unlabeled examples, so the sums inside each likelihood calculation have a different number of terms. This makes it difficult to choose values of α in an interpretable way. Choosing $\alpha = 0.6$ and $\beta = 0.4$ does not correspond to maximizing with 60% of the weight on the discriminative component, as the discriminative gradient magnitudes tend to be larger than those of the generative component.

One potential solution to this problem is to normalize each of the components so that they have the same magnitude, and weight the normalized componenets. In non-log space, normalizing each component corresponds to raising each component to a power x. If $x > 1$, then this makes the probability distribution more peaked, whereas if $x < 1$, the probability distribution is flattened. Since $P(Y_L|X_L;\Theta)$ is convex, stretching or flattening it should make little difference in terms of the ability of a gradient-based optimizer to find the maximum. However, $P(X;\Theta)$ is not convex, and consequently flattening it could actually change the maximum found by the maximizer, if x is small enough to sufficiently smooth the distribution. Because the generative likelihood is smaller in magnitude and flattening it by raising it to a power $x < 1$ may be detrimental, we avoid normalization and set $\beta = 1$ and $\alpha >> \beta$.

The difference in the magnitude of the likelihoods could also cause maximization to appear to converge when one component conceals the changes in the other. To deal with this issue, we adapt convergence criteria so that training converges only when both components, considered independently, have converged.

In addition to the terms above, we use a standard zero-mean Gaussian prior over parameters:

$$\ln P(\Theta) \propto -\frac{||\Theta||^2}{\sigma^2}.$$

2.2 Parameter Coupling Prior

In the approach of Lasserre, et al. [11], which again we refer to as the *parameter coupling prior* approach, the generative and discriminative components have separate sets of parameters. The two sets of parameters are jointly trained and are coupled using a prior distribution. Following [11], we define the joint distribution of features X, classes Y, and parameters Θ_D and Θ_G (for the discriminative and generative models, respectively) as:

$$P(X, Y, \Theta_D, \Theta_G) = P(\Theta_D, \Theta_G)P(Y|X;\Theta_D)P(X;\Theta_G)$$

Let us consider two special cases of priors. If the prior $P(\Theta_D, \Theta_G)$ constrains $\Theta_D = \Theta_G$, then we have a generative model based on the joint distribution.

$$P(X, Y, \Theta_G) = P(\Theta_G)P(X, Y;\Theta_G)$$

If the prior assumes that the two sets of parameters are independent $P(\Theta_D, \Theta_G) = P(\Theta_D)P(\Theta_G)$, then we have:

$$P(X, Y, \Theta_D, \Theta_G) = P(\Theta_D)P(Y|X;\Theta_D)\,[P(\Theta_G)P(X;\Theta_G)]$$

In other words, if the underlying joint model is such that the parameters of the marginal model and the conditional model are completely independent, then the terms inside the brackets are constant with respect to Θ_D, and hence play no role in classification. Therefore, this is a discriminative model.

A prior that imposes a soft constraint that the parameters must be similar allows blending of the generative and discriminative. As in [11], we couple the two sets of parameters by a prior of the form:

$$\ln P(\Theta_D, \Theta_G) = -\frac{||\Theta_D - \Theta_G||^2}{2\sigma^2} \quad (2)$$

Lasserre et al. noted that the generative component $P(X;\Theta_G)$ can make use of unlabeled data, and can hence be used for semi-supervised learning. Experimental results on one dataset using the above prior demonstrated the potential for semi-supervised learning using this method.

3. MODEL

It is important to note that the two approaches described above are model-independent because they only specify the form of the objective function. We can derive concrete versions of the objective functions for a specific graphical model. Here, we apply them to a Markov random field

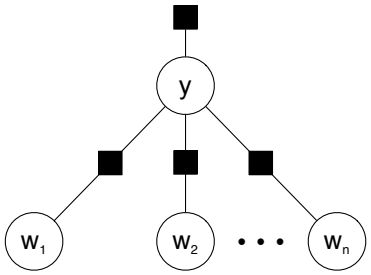

Figure 1: A factor graph for a naively-structured Makrov random field model.

(MRF) model, an undirected graphical model. The model is structured so that the input variables $\mathbf{w}_i$ are conditionally independent given the class $\mathbf{y}$. This can be interpreted as an undirected analog to naive Bayes models. For this reason we refer to this specific structure as a naively structured MRF. The factor graph for this model is shown in Figure 1. We could also use these training objective functions in more complicated models with hidden topic variables or models for sequences.

3.1 Training

We estimate the parameters of the model by finding the parameters that maximize the objective functions defined in Section 2. We use gradient methods to find the maximum. Below we define these objective functions for the naively-structed MRF model, and compute the gradients. The components of each objective are derived from the joint distribution of the model given by:

$$P(\mathbf{y}, \mathbf{x}) = \frac{1}{Z}\exp(\mathbf{\Theta}_y^T \mathbf{y} + \mathbf{y}^T \mathbf{\Theta}_{xy}^T \mathbf{x}) \tag{3}$$

Where $Z = \sum_{\mathbf{y}} \sum_{\mathbf{x}} \exp(\mathbf{\Theta}_y^T \mathbf{y} + \mathbf{y}^T \mathbf{\Theta}_{xy}^T \mathbf{x})$ is a normalizing factor that ensures a true probability distribution. First, we derive the gradient for the discriminative component of the objective function, the conditional log-likelihood $L_{\mathbf{y}|\mathbf{x}} = \log P(\tilde{\mathbf{y}}|\tilde{\mathbf{x}})$, where $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{x}}$ denote observations and

$$L_{\mathbf{y}|\mathbf{x}} = \sum_{(\tilde{\mathbf{x}},\tilde{\mathbf{y}})\in D_L} \left(\mathbf{\Theta}_y^T \tilde{\mathbf{y}} + \tilde{\mathbf{y}}^T \mathbf{\Theta}_{xy}^T \tilde{\mathbf{x}} - \ln Z(\tilde{\mathbf{x}})\right),$$

where

$$Z(\tilde{\mathbf{x}}) = \sum_{\mathbf{y}} \exp(\mathbf{\Theta}_y \mathbf{y} + \mathbf{y}^T \mathbf{\Theta}_{xy}^T \tilde{\mathbf{x}}).$$

The gradient is then computed as

$$\begin{aligned}\frac{\partial L_{\mathbf{y}|\mathbf{x}}}{\partial \mathbf{\Theta}_{xy}} &= \sum_{(\tilde{\mathbf{x}},\tilde{\mathbf{y}})\in D_L} \left(\tilde{\mathbf{x}}\tilde{\mathbf{y}}^T - \frac{\sum_{\mathbf{y}} \exp(\mathbf{\Theta}_y \mathbf{y} + \mathbf{y}^T \mathbf{\Theta}_{xy}^T \tilde{\mathbf{x}}) \tilde{\mathbf{x}} \mathbf{y}^T}{\sum_{\mathbf{y}} \exp(\mathbf{\Theta}_y \mathbf{y} + \mathbf{y}^T \mathbf{\Theta}_{xy}^T \tilde{\mathbf{x}})}\right) \\ &\propto E_{\tilde{\mathbf{x}},\tilde{\mathbf{y}}}[\mathbf{x}\mathbf{y}^T] - E_{\tilde{\mathbf{x}}}\left[E_{\mathbf{y}|\mathbf{x}}[\mathbf{x}\mathbf{y}^T]\right]\end{aligned}$$

where $E_{\mathbf{x}}[\cdot]$ denotes the expectation under $p(\mathbf{x})$ and $E_{\tilde{\mathbf{x}}}[\cdot]$ denotes an expectation under the empirical distribution of $\mathbf{x}$. The gradient of $L_{\mathbf{y}|\mathbf{x}}$ with respect to parameters $\mathbf{\Theta}_y$ is similar. The log marginal likelihood for $P(\mathbf{x})$ is

$$L_{\mathbf{x}} = \sum_{(\tilde{\mathbf{x}})\in D} \left(\ln \sum_{\mathbf{y}} (\exp(\mathbf{\Theta}_y \mathbf{y} + \mathbf{y}^T \mathbf{\Theta}_{xy}^T \tilde{\mathbf{x}})) - \ln Z\right)$$

and has gradient

$$\begin{aligned}&\frac{\partial}{\partial \mathbf{\Theta}_{xy}} \sum_{(\tilde{\mathbf{x}})\in D_U} \left(\ln \sum_{\mathbf{y}} (\exp(\mathbf{\Theta}_y \mathbf{y} + \mathbf{y}^T \mathbf{\Theta}_{xy}^T \tilde{\mathbf{x}})) - \ln Z\right) \\ &= \sum_{(\tilde{\mathbf{x}})\in D_U} \left(\frac{\sum_{\mathbf{y}} (\exp(\mathbf{\Theta}_y \mathbf{y} + \mathbf{y}^T \mathbf{\Theta}_{xy}^T \tilde{\mathbf{x}})) \tilde{\mathbf{x}} \mathbf{y}^T}{\sum_{\mathbf{y}} (\exp(\mathbf{\Theta}_y \mathbf{y} + \mathbf{y}^T \mathbf{\Theta}_{xy}^T \tilde{\mathbf{x}}))} - \frac{\partial \ln Z}{\partial \mathbf{\Theta}_{xy}}\right) \\ &\propto E_{\tilde{\mathbf{x}}}\left[E_{\mathbf{y}|\mathbf{x}}[\mathbf{x}\mathbf{y}^T]\right] - E_{\mathbf{x},\mathbf{y}}[\mathbf{x}\mathbf{y}^T]\end{aligned}$$

For the multi-conditional objective function, the parameter matrices $\mathbf{\Theta}_{xy}$ and $\mathbf{\Theta}_y$ are the same for both the discriminative and generative components. For the parameter coupling prior method, $\mathbf{\Theta}_{xy}$ and $\mathbf{\Theta}_y$ are different for each component and are coupled by (2). The derivative of this prior with respect to each set of parameters is:

$$\begin{aligned}\ln P(\mathbf{\Theta}_D, \mathbf{\Theta}_G) &\propto -\frac{||\mathbf{\Theta}_D - \mathbf{\Theta}_\mathbf{G}||^2}{2\sigma^2} \\ \frac{\partial}{\partial \mathbf{\Theta}_D} \ln P(\mathbf{\Theta}_D, \mathbf{\Theta}_G) &= -\frac{\mathbf{\Theta}_D - \mathbf{\Theta}_G}{\sigma^2} \\ \frac{\partial}{\partial \mathbf{\Theta}_G} \ln P(\mathbf{\Theta}_D, \mathbf{\Theta}_G) &= -\frac{\mathbf{\Theta}_G - \mathbf{\Theta}_D}{\sigma^2}\end{aligned}$$

If $\beta = 0$ in the multi-conditional approach or $\sigma = \infty$ in the parameter coupling priors approach, then the L_x term drops out of the objective function and we only maximize $L_{y|x}$ which uses no unlabeled data. Maximizing only $L_{y|x}$ in the naive MRF model yields a maximum entropy classifier [16]. We use maximum entropy classifiers as the supervised counterpart to our hybrid semi-supervised methods.

We use Limited-Memory-BFGS, a quasi-Newton optimization method that has been shown to work well for maximum entropy models [12], in conjunction with the adapted converge criteria discussed in Section 2.1. Note that the marginal density $P(\mathbf{x})$ is not convex, which means neither the multi-conditional nor the parameter coupling prior objective functions are convex. Because L-BFGS is a convex optimizer, we converge to a local maximum. Empirically, we have found that L-BFGS requires fewer training iterations converges to better maxima than other convex optimizers. Using a method to specifically address the lack of convexity may be beneficial, but is an issue we leave for future research.

4. RELATED WORK

4.1 Semi-Supervised Learning

Although many semi-supervised methods have been proposed, they each fall into one of a small number of classes of methods, each of which make certain assumptions.

Co-training [2] and related multi-view learning methods [4, 20] assume that multiple classifiers are trained over multiple feature views (splits) of the same labeled examples. As capacity control, these classifiers are encouraged to make the

same prediction on any unlabeled example. However, multiple feature views often do not naturally exist in practice, and these methods resort to artificially creating random feature splits.

Graph based semi-supervised learning [22, 1] assumes that labeled and unlabeled examples are connected by a graph, where edges represent similarity between examples. The discriminant function is encouraged to vary smoothly with respect to the graph. As a result, connected nodes tend to have the same label. One interpretation of this it is that labels 'propagate' through unlabeled examples via graph edges. However, the graph is usually constructed from the distances in feature space, and is susceptible to overlapping classes. Indeed if unlabeled data from different classes strongly overlap, the graph will be wrong, and the method can be expected to be inferior to supervised learning.

Semi-supervised or transductive support vector machines (S3VMs, TSVMs) [8, 7] also assume that there is a wide margin in kernel induced feature space between unlabeled data from different classes. The margin may be different than the traditional margin of labeled examples. S3VMs attempt to place the decision boundary in the unlabeled margin. However, there are two issues: First, such margin may not exist if the classes strongly overlap even in the kernel induced feature space. Second, S3VMs involves a highly non-convex optimization problem which is difficult to solve [6].

Generative semi-supervised methods based on the expectation maximization (EM) algorithm assume a model for the input distribution. In [17] Nigam, et al. use the EM algorithm in conjunction with a mixture of multinomials, or naive Bayes, generative model. In the E step, each unlabeled example is assigned a label distribution according to its expected value under the current model. In the M step, the multinomial parameters are re-estimated. In practice this model can fail when the assumption of independent input features is violated or when the best generative structure does not correspond to the decision boundary.

4.2 Generative Discriminative Hybrids

There have been several successful applications of hybrid generative/discriminative methods in addition to the two approaches that are the focus of this paper. In many approaches, parameters are separated into two subsets, one of which is trained discriminatively and the other generatively.

Raina et al. [18], present a model for document classification in which documents are split into multiple regions. For a newsgroup message, regions might include the header and the body. In this model, each region has its own set parameters that are trained generatively, while the parameters that weight the importance of each region in the final classification are trained discriminatively. Experimental results show that this hybrid algorithm gives better classification accuracy than either naive Bayes or logistic regression (a generative/discriminative pair [15]) alone. Additionally, Raina, et al. show that because the number of discriminative parameters in the model is small, only a small amount of training data is required to estimate these parameters.

Kang and Tian [9] extend naive Bayes by splitting features into two sets X_1 and X_2. The directed graphical model for this approach has nodes X_1 as the parents of the class variable Y, and parameters X_2 as the children of Y. Parameters of this model are estimated by maximizing $P(Y|X_1)$ (the discriminative component), and $P(X_2|Y)$ (the generative component). An iterative algorithm based on classification accuracy is used to decide which features go in X_1.

Bouchard and Triggs [3] propose a method to trade-off generative and discriminative modeling that is similar to multi-conditional learning because it maximizes a weighted combination of two likelihood terms using one set of parameters. Defining $L_{gen} = \ln P(Y, X)$ and $L_{disc} = \ln P(Y|X)$, Bouchard and Triggs present a combined objective function $L_\lambda = \lambda \ln P(X, Y) + (1 - \lambda) \ln P(Y|X)$. A subtle difference between this objective function and those presented here is that in their approach the generative component is the full joint distribution. As in other related work, experimental results show that highest accuracy is obtained somewhere between fully generative (in this case $\lambda = 1$) and fully discriminative ($\lambda = 0$).

5. EXPERIMENTS

Semi-supervised learning methods are rarely applied in practice, in part because there are few empirical comparisons of multiple semi-supervised methods. Additionally, many semi-supervised experiments use binary datasets that have few features and are easily separable. Here we provide a substantial comparison of two semi-supervised hybrid generative/discriminative methods and two prominent semi-supervised learning methods, as well as their supervised counterparts, on multi-class datasets with large numbers of features.

5.1 Setup

We first discuss the supervised/semi-supervised pairs of methods we use in the experiments.

Naive Bayes / EM Naive Bayes (*nb,emnb*)
We use the naive Bayes implementation in the Mallet toolkit [14]. As in Nigam et.al [17], we use Laplace (plus-1) smoothing, so that unseen events do not get zero probability.

SVM / Transductive SVM (*svm,tsvm*)
In our experiments we use Universvm, an SVM implementation introduced in [7] that uses the Concave-Convex Procedure (CCCP) to optimize transductive SVMs. Collobert et al. show that optimizing TSVMs with CCCP improves accuracy and decreases training time when compared to other heuristic methods. The TSVM introduces several hyperparameters that need to be tuned. In our experiments, we tune $C \in \{10^{-5}, 10^{-3}, 10^{-1}, 10, 10^3, 10^5\}$ and $C^* \in \{10^{-5}, 10^{-3}, 10^{-1}, 0, 10\}$, the cost parameters for the labeled and unlabeled data, respectively. Collobert et al. also tune the symmetric ramp loss for the unlabeled data, but doing a grid search over three parameters, each with several possible values, was not practical for large-scale comparison. We set the symmetric ramp loss parameter to -0.5 in all experiments and use a linear kernel. Although we do not perform normalization of feature counts for other methods, we find that this is very important to achieve reason-

able TSVM results and therefore normalize feature vectors to have Euclidean length 1 for the SVM and TSVM experiments.

Max Entropy / Multi-Conditional Method (*me*,*mcl*)
We use the implementation of maximum entropy models in Mallet [14] and also use this framework to implement the multi-conditional method. For both we tune the Gaussian prior variance $\sigma^2 \in \{0.01, 0.1, 1, 10, 100\}$. For the multi-conditional method, we also tune the relative weighting of the discriminative component $\alpha \in [10^2, 10^7]$ at every order of magnitude. We use $\beta = 1$ for all experiments. See Section 2.1 for some discussion of settings for α and β.

Parameter Coupling Prior (*pcp*)
We implement this model in Mallet [14] as well. Following Lasserre et al. [11], we tune the parameter α, which is translated into a value for σ, the strength of the coupling prior, using $\sigma(\alpha) = (\frac{\alpha}{1-\alpha})^2$. We use $\alpha = [0.1, 0.9]$ in intervals of 0.1. We use $\sigma^2 = 1$ for the Gaussian prior on parameters, because we find that tuning this value provides little benefit when compared to the extra time it requires (we later show that *pcp* requires more time to train than the other methods). As above, the supervised counterpart for this method is the maximum entropy classifier.

We run experiments on five text classification datasets and one sliding-window sequence labeling classification dataset. For the text classification datasets, features correspond to word occurrence counts. For the NER task, features are binary word occurrences and properties of those words (such as capitalization) within three time steps. Stopwords, HTML, message headers (where appropriate), and features that only occurred once are removed from all datasets. Where noted, low frequency features are also removed.

Datasets

- *movie* (24,841 features, 2 classes) Classify the sentiment of movie reviews from IMDB as *positive* or *negative*.
- *webkb* (22,824 features, 4 classes) Classify university webpages as *student*, *course*, *faculty*, or *project*.
- *sraa* (77,494 features, 4 labels) Classify messages by the newsgroup to which they were posted: *simulated-aviation*, *real-aviation*, *simulated-autoracing*, *realauto*.
- *sector* (22,835 features 38 labels) Classify webpages into specific industry sectors.
- *blogs* (95,583 features, 4 labels) Classify the age of a blogger given blog posts. Due to the large number of features in this dataset, those that occur less than 10 times are removed. This dataset was introduced in [19].
- *ner* (60,502 features, 9 labels) Sliding-window named-entity recognition using the CoNLL 2003 dataset.

Although all datasets are labeled, we simulate unlabeled data by ignoring the labels for some examples. Specifically, we choose examples to remain labeled randomly, but ensure that the number of labeled examples is the same for each class as in [17]. We treat the remaining examples as unlabeled, up to a maximum of 5,000 unlabeled examples. The success of semi-supervised learning is dependent on the quality of the "seed" set of labeled examples. Therefore, we average results over five random labeled sets. We report accuracy on a held-out test data, rather than reporting accuracy on the unlabeled data, as is done with TSVMs [7].

Many semi-supervised methods introduce hyperparameters including graph methods [22, 1] and TSVMs [7]. We discuss the issue of choosing hyperparameters and provide some heuristics that reduce the need for hyperparameter tuning for generative/discriminative methods in Section 5.3. For these experiments we use a grid search to find the best settings, and use parameters settings that give the best test set accuracy, as also in [7]. Therefore, these results can therefore be interpreted as an indication of the potential of these methods, though further research is needed for practical parameter tuning.

5.2 Results and Discussion

Classification accuracy results are presented in Table 1. Either MCL or PCP achieves the highest accuracy in 75% of the experiments, and MCL and PCP achieve the largest mean accuracy improvements over their supervised counterpart at 5.2% and 3.1%, respectively. Additionally, MCL is the only method to show semi-supervised improvements on every dataset. Figure 2 illustrates that for both MCL and PCP the best accuracies are attained in the space between purely generative and purely discriminative.

We argue in the introduction that hybrid approaches are able to avoid or mitigate the assumptions of other semi-supervised methods. We would like to determine if the cases in which the hybrid methods perform well empirically match our intuitive justifications. It is difficult to quantifiably compare the degree of overlapping classes or the degree to which model assumptions fail across text datasets, but we can gain some insight by considering the datasets on which hybrid semi-supervised methods do well and other methods do poorly.

First notice that naive Bayes performs worst on datasets *sector* and *ner*, in which EM naive Bayes gives lower accuracy than supervised naive Bayes. Additionally, despite semi-supervised improvements, EM naive Bayes gives much lower accuracy on *blogs* than other methods. These three datasets have the largest number of classes, most complicated and correlated features, and greatest number of features, respectively. Highly correlated features clearly violate the "naive" assumption of feature independence. We also expect that for these tasks the natural clusters in the data will not necessarily be correlated with the decision boundary. For example, both *ner* and *blog* involve a very specific task on text that includes a wide variety of words and topics. The hybrid methods are able to mitigate these generative assumptions using the discriminative component.

Relative to the other methods, TSVMs perform poorly on *ner* and *sraa*, cases in which we expect classes to be strongly overlapping. The *sraa* dataset contains messages from newsgroups on simulated aviation, real aviation, simulated auto

dataset	nb	emnb	svm	tsvm	me	mcl	pcp
movie (10)	59.7	62.1	57.9	58.5	58.0	59.0	**64.6**
movie (25)	62.0	61.2	63.5	62.6	63.0	64.0	**68.6**
webkb (10)	60.4	65.6	66.8	66.2	64.9	67.6	**72.5**
webkb (25)	68.2	73.4	76.3	75.9	76.5	**76.9**	76.7
sector (5)	30.7	13.2	42.7	**45.5**	35.8	37.6	28.1
sector (10)	41.2	21.6	54.9	**56.8**	48.6	50.4	42.7
ner (10)	26.1	2.6	87.4	26.9	77.7	**88.6**	75.8
ner (25)	25.4	3.0	**90.1**	27.7	84.6	**90.1**	84.3
sraa (10)	63.1	70.6	64.3	67.6	61.9	72.7	**81.6**
sraa (25)	74.5	78.5	71.6	74.3	69.3	78.6	**84.1**
blogs (10)	30.6	38.8	47.5	48.0	42.2	**49.8**	41.1
blogs (25)	32.3	35.7	52.6	53.7	48.1	**57.6**	47.8
mean	47.9	43.9	64.6	55.3	60.9	**66.1**	64.0

Table 1: Classification accuracy results. Parenthesized values indicate the number of labeled documents per class.

feature	diff	me	mcl	feature	diff	me	mcl
im	.039	.259	.298	nbsp	-.022	.237	.215
dont	.021	.255	.277	urllink	-.012	.246	.234
lol	.020	.258	.278	arianna	-.007	.249	.243
today	.020	.271	.291	years	-.007	.240	.234
haha	.019	.252	.271	wife	-.006	.242	.235
yeah	.016	.254	.270	frienz	-.005	.261	.256
good	.016	.259	.275	couple	-.005	.259	.254
fun	.014	.260	.273	moved	-.005	.245	.239
school	.013	.263	.276	town	-.005	.252	.248
pretty	.010	.254	.264	bar	-.004	.247	.243
home	.010	.248	.258	renee	-.004	.245	.241
gonna	.010	.259	.269	world	-.004	.249	.244
day	.010	.273	.283	doesn	-.004	.256	.251
kinda	.009	.251	.261	city	-.004	.253	.249
guess	.009	.249	.258	ago	-.004	.254	.250

Table 2: Parameters with largest positive and negative differences in discriminative power for the *blog* dataset with 10 labeled documents per class for label *age 10-19* between supervised and semi-supervised with the multi-conditional method. The Gaussian prior variance for the maximum entropy classifier is 1.0. For the MCL method, $\alpha = 10^5$ and the Gaussian prior variance is 1.0, corresponding to a case when the testing accuracy was 55.6.

feature	diff	me	pcp	feature	diff	me	pcp
airspeed	.122	.252	.374	font	-.092	.250	.158
altitude	.109	.280	.390	autos	-.089	.249	.160
pitch	.103	.252	.355	div	-.079	.250	.171
preflight	.103	.263	.366	nextpart	-.069	.250	.181
rudder	.103	.284	.387	iso	-.066	.250	.184
stephenames	.102	.261	.362	voodoo	-.066	.244	.178
downwind	.101	.251	.352	meta	-.064	.250	.186
ppl	.100	.282	.381	px	-.063	.250	.187
ames	.098	.267	.365	printable	-.063	.250	.187
prop	.098	.278	.375	mb	-.062	.246	.184
garden	.096	.257	.353	racing	-.061	.233	.172
afm	.095	.250	.345	gb	-.061	.250	.189
flew	.095	.251	.346	ns	-.061	.248	.187
crab	.095	.250	.345	cars	-.061	.238	.177
climb	.094	.251	.345	version	-.006	.232	.172

Table 3: Parameters with largest positive and negative differences in discriminative power for the *sraa* dataset with 25 labeled documents per class for label *real aviation* between supervised and semi-supervised with the parameter coupling prior method. The Gaussian prior variance for the maximum entropy classifier is 0.1. For the PCP method, $\alpha = 0.6$, corresponding to a case when the testing accuracy was 87.5.

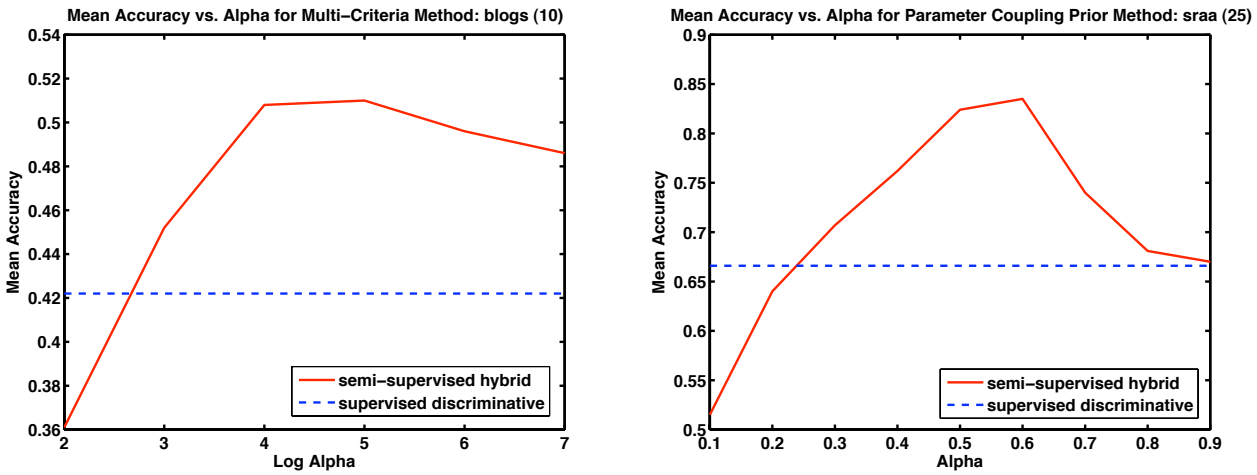

Figure 2: Mean accuracies vs. α for hybrid methods. The dashed line represents the supervised accuracy. The best accuracies are obtained in the region between purely generative and purely discriminative.

racing, and real automobiles. These classes have high word overlap, as both simulated and real automobile newsgroups will include words like "tire" and "engine", for example. The *ner* dataset contains classes for the start of a token type, such as location, and the continuation of a token type. We expect that the "begin" and "continuation" classes for a token type will not have a low density region between them, and we see in the results that the TSVM chooses poor locations for the decision boundaries as the accuracy is much worse than supervised.

We note that we originally conducted the SVM/TSVM experiments without normalizing feature vectors, and the TSVM only improved accuracy over the supervised SVM in two cases (blog (10,25)). Interestingly, normalization seems to have a much larger impact on semi-supervised learning than on supervised learning. Excluding the *ner* dataset, the mean SVM accuracy increases by 1.8% using normalization, whereas the mean TSVM accuracy increases by 5.2%. Even when using normalization, TSVMs do not always give improvements over supervised SVMs. One possible explanation is that most published evaluations of TSVMs use simple, two-class datasets. Additionally, TSVM experiments typically use the eventual test data as unlabeled data during training, whereas here we use a held-out test set that is separate from the unlabeled training data. A more complicated kernel may improve accuracy, but note that the other methods only use linear combinations of features.

Comparing the two hybrid generative/discriminative methods, MCL achieves higher mean accuracy than PCP on seven of the twelve experiments, as well as higher overall mean accuracy across all tasks. Notice that the two methods rarely perform well on the same dataset. The fundamental difference between the two methods is that MCL has one shared set of parameters while PCP has two coupled sets of parameters. An advantage of the PCP method is that the gradients of the two components do not directly compete to modify parameter values. This allows each component of the PCP objective to have more freedom in modeling the data. If some generative parameter needs to be set in a way which does not correspond to a good discriminative setting, this penalty can absorbed if it leads to an improved model. Additionally, the PCP method allows the inclusion of a prior that only effects the discriminative parameters. With MCL, a Gaussian prior on parameters helps prevent the discriminative component from overfitting, but because the parameters are shared, this prior has a negative impact on the generative component. Namely, we observe that the penalty on large parameter values tends to prevent large changes in parameter value due to the generative component, as empirically the generative parameters need to be more spread out than their discriminative counterparts. However, it appears that the extra modeling power afforded by the PCP method sometimes makes it more difficult to find good parameter settings during training. We note that the cases in which MCL gives improved accuracy over PCP tend to be more complicated datasets in terms of the number of labels or features. This suggests that PCP may be preferable for less complicated datasets and MCL for more complicated datasets.

In Tables 2 and 3 we show how parameter values change after introducing the generative component that uses unlabeled data. These tables show differences in the discriminative power of features in a supervised maximum entropy model and a semi-supervised hybrid. The values for each feature can be interpreted as the probability that a single word document containing that word feature belongs to the class listed in the table caption. In Table 2, we see that for the *blogs* dataset and label *age 10-19*, the discriminative power of "im", "lol", "school" and "home" increases, and the discriminative power of "wife", "couple", and "bar" decreases. In Table 3, we see that for dataset *sraa* and label *real aviation*, the discriminative power of "airspeed", "altitude", and "preflight" increases, while the power of "autos", "racing", and "cars" decreases. Intuitively, we see that the addition of the generative component helps to boost the discriminative power of features that the supervised model could not discover with limited labeled data.

method	movie (10)	sraa (10)	ner (10)
emnb	0.8	2.3	4.8
svm	0.8	8.5	63.4
tsvm	30.9	796.4	1052.4
me	2.0	12.2	25.8
mcl	41.8	745.1	1535.3
pcp	124.2	3090.4	2885.2
pcp-init	70.1	1618.3	2182.6

Table 4: Mean training time in seconds.

5.3 Practical Considerations

Hyperparameter values are often important to the success of semi-supervised learning methods, but tuning hyperparameters can be difficult in practice. In supervised learning, hyperparameter values are typically chosen using cross-validation. If we apply this method for semi-supervised learning, each fold test set ends up with a very small number of labeled examples, since the total number of labeled examples is small. This means that the performance estimate obtained from this test set may be inaccurate. Additionally, optimal hyperparameter values may depend upon the specific set of labeled examples. Training with a subset of those labeled examples could result in drastically different ideal settings for the the hyper-parameters, as, for example, a poor seed set of labeled examples may require a hybrid classifier to put more weight on the generative component and rely on unlabeled data. Another option is to choose the hyperparameters that give the highest likelihood on the training data. However, we have found that these hyperparameters produce the highest likelihood models because of overfitting. Finally, choosing hyperparameters using a validation set is not practical, as if more labeled data is available, it would almost certainly be more beneficial to use that data during training than to use it to tune hyperparameters.

Another practical issue in semi-supervised learning is that the added complexity of semi-supervised training algorithms increases the overall model training time, as shown in Table 4. On average, the PCP method takes the longest to train. In addition to having twice as many parameters as the MCL method, it also seem to take more iterations to converge. The EM naive Bayes semi-supervised learning is extremely fast because the maximum likelihood estimates can be computed in closed form.

We propose a heuristic for hybrid models to simultaneously reduce training time and reliance on hyperparameter settings that involves ensuring reasonable initial parameter settings. For the PCP method, we initialize both sets of parameters to the results of supervised training on the labeled data. We present results using supervised initialization of the PCP method in Table 5. This also reduces training time, as shown by the entry *pcp-init* in Table 4. For the MCL method, we propose to use the discriminative component exclusively in the first few iterations of training. Since the discriminative component is convex, this guarantees that we move into a reasonable region parameter space. At the same time, by avoiding going the whole way to the global maximum, we give the parameters room to move away from the discriminative maximum.

dataset	accuracy
movie (10)	59.4
movie (25)	67.7
webkb (10)	67.9
webkb (25)	78.4
sector (5)	37.9
sector (10)	51.0
ner (10)	75.6
ner (25)	84.5
sraa (10)	72.3
sraa (25)	77.8
blogs (10)	46.0
blogs (25)	49.3

Table 5: Classification results when using supervised initialization for parameter coupling prior model. Parenthesized values indicate the number of labeled documents per class.

These heuristics seem to make the algorithms more consistent, as the mean accuracy across all hyperparameter settings is higher. However, the maximum accuracies attained are sometimes lower. Intuitively, one of the real benefits of these methods is that the generative component can pull the discriminative component into regions of parameter space very different from those chosen by supervised discriminative learning on the limited labeled data. With these initializations, we ensure that we do not converge to a poor local maximum, but sacrifice the potential to find drastically different parameter settings.

6. CONCLUSION AND FUTURE WORK

We have considered hybrid generative/discriminative approaches to semi-supervised classification in which the generative component includes unlabeled data. We compare two methods for combining generative and discriminative likelihood in detail: a multi-conditional learning method and a method where each component has its own set of parameters that are coupled by a prior distribution. In a substantial empirical comparison, a hybrid method provides the best accuracy in eight of the twelve experiments. Intuitively, we conjecture that they perform well by mitigating the modeling assumptions of generative semi-supervised methods and avoiding the low-density-between-class assumptions of discriminative semi-supervised methods.

In future work, we would like to apply hybrid generative/discriminative approaches to transfer or multi-task learning, consider heuristic and probabilistic methods to learn hyperparameters from data, and research alternative optimization techniques utilizing ideas from the multi-criteria optimization literature.

7. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249, in part by NSF Nano #DMI-0531171, and in part by the Defense Advanced Research Projects Agency (DARPA), through the

Department of the Interior, NBC, Acquisition Services Division, under contract number NBCHD030010. CP appreciates support by Microsoft Research under the Memex and eScience funding programs and support from Kodak Research. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

8. REFERENCES

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from examples. Technical Report TR-2004-06, University of Chicago, 2004.

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, 1998.

[3] G. Bouchard and B. Triggs. The tradeoff between generative and discriminative classifiers. In J. Antoch, editor, *Proceedings in Computational Statistics, 16th Symposium of IASC*, volume 16, Prague. Physica-Verlag.

[4] U. Brefeld, T. Gaertner, T. Scheffer, and S. Wrobel. Efficient co-regularized least squares regression. In *ICML06, 23rd International Conference on Machine Learning*, Pittsburgh, USA, 2006.

[5] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[6] O. Chapelle, V. Sindhwani, and S. S. Keerthi. Branch and bound for semi-supervised support vector machines. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

[7] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive SVMs. *The Journal of Machine Learning Research*, 7(Aug):1687–1712, 2006.

[8] T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. 16th International Conf. on Machine Learning*, pages 200–209. Morgan Kaufmann, San Francisco, CA, 1999.

[9] C. Kang and J. Tian. A hybrid generative/discriminative bayesian classifier. In *Proceedings of the 19th International FLAIRS Conference*, 2006.

[10] B. M. Kelm, C. Pal, and A. McCallum. Combining generative and discriminative methods for pixel classification with multi-conditional learning. *ICPR*, 2:828–832, 2006.

[11] J. A. Lasserre, C. M. Bishop, and T. P. Minka. Principled hybrids of generative and discriminative models. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 87–94, Washington, DC, USA, 2006. IEEE Computer Society.

[12] R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *In Sixth Conf. on Natural Language Learning*, pages 49–55, 2002.

[13] A. McCallum, C. Pal, G. Druck, and X. Wang. Multi-conditional learning: Generative/discriminative training for clustering and classification. In *AAAI '06: American Association for Artificial Intelligence National Conference on Artificial Intelligence*, 2006.

[14] A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[15] A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[16] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification, 1999.

[17] K. Nigam, A. McCallum, S. Thrun, and T. M. Mitchell. Learning to classify text from labeled and unlabeled documents. In *AAAI/IAAI*, page 792, 1998.

[18] R. Raina, Y. Shen, A. Y. Ng, and A. McCallum. Classification with hybrid generative/discriminative models. In *NIPS*, 2003.

[19] J. Schler, M. Koppel, S. Argamon, and J. Pennebaker. Effects of age and gender on blogging. In *2006 AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs*, 2006.

[20] V. Sindhwani, P. Niyogi, and M. Belkin. A co-regularized approach to semi-supervised learning with multiple views. In *Proc. of the 22nd ICML Workshop on Learning with Multiple Views*, August 2005.

[21] D.-Q. Zhang and S.-F. Chang. A generative-discriminative hybrid method for multi-view object detection. *CVPR*, 2:2017–2024, 2006.

[22] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML-03, 20th International Conference on Machine Learning*, 2003.