Semi-supervised Learning of Classifiers: Theory, Algorithms for Bayesian Network Classifiers and Application to Human-Computer Interaction

Ira Cohen, Fabio G. Cozman, Nicu Sebe, Marcelo C. Cirelo, Thomas S. Huang

Abstract

Automatic classification by machines is one of the basic tasks required in any pattern recognition and human computer interaction applications. In this paper we discuss training probabilistic classifiers with labeled and unlabeled data. We provide a new analysis which shows under what conditions unlabeled data can be used in learning to improve classification performance. We also show that if the conditions are violated, using unlabeled data can be detrimental to classification performance. We discuss the implications of this analysis to a specific type of probabilistic classifiers, Bayesian networks, and propose a new structure learning algorithm that can utilize unlabeled data to improve classification. Finally, we show how the resulting algorithms are successfully employed in two applications related to human-computer interaction and pattern recognition; facial expression recognition and face detection.

Keywords

Semi-supervised learning, generative models, facial expression recognition, face detection, unlabeled data, Bayesian network classifiers.

I. Introduction

Many pattern recognition and human computer interaction applications require the design of classifiers. Classifiers are either designed from expert knowledge or using training data. Training data can be either labeled to the different classes or unlabeled. In many applications, obtaining fully labeled training sets is a difficult task; labeling is usually done using human expertise, which is expensive, time consuming and error prone. Obtaining unlabeled data is usually easier since it involves collecting data that is known to belong to one of the classes without having to label it, e.g., in facial expression recognition, it is easy to collect videos

Ira Cohen and Thomas S. Huang: Beckman Institute, University of Illinois at Urbana-Champaign, USA. {iracohen,huang}@ifp.uiuc.edu

Fabio G. Cozman and Marcelo C. Cirelo: Escola Politécnica, Universidade de São Paulo, São Paulo,Brazil. fgcozman@usp.br, marcelo.cirelo@poli.usp.br

Nicu Sebe: Faculty of Science, University of Amsterdam, The Netherlands. nicu@science.uva.nl

of people displaying expressions, but it is very tedious and difficult to label the video to the corresponding expressions. Learning with both labeled and unlabeled data is known as semi-supervised learning.

We start with a general analysis of semi-supervised learning for probabilistic classifiers. The goal of the analysis is to show under what conditions unlabeled data can be used to improve the classification accuracy. We review maximum likelihood estimation when learning with labeled and unlabeled data. We provide an asymptotic analysis of the value of unlabeled data under this framework to show that unlabeled data helps in reduce the estimator's variance. We show that when the assumed probabilistic model matches the true data generating distribution, the reduction in variance leads to an improved classification accuracy; which is not surprising, and has been analyzed before [1, 2]. However, we show how when the assumed probabilistic model does not match the true data generating distribution, using unlabeled data can be detrimental to the classification accuracy; a result that was generally ignored or misinterpreted by previous researchers who observed it empirically before [1, 3, 4].

This new result emphasizes the importance of using correct modeling assumption when learning with unlabeled data. As classifiers in our applications, we choose Bayesian networks. For Bayesian network classifiers, our analysis of semi-supervised learning implies a need to find a structure of the graph that matches the true distribution generating the data. What we try to emphasize is that, while in many classification problems simple structures learned with just labeled data have been used successfully (e.g., the Naive-Bayes classifier [5, 6]), such structures failed when trained with both labeled and unlabeled data [7].

Bayesian networks are probabilistic classifiers, in which the joint distribution of the features and class variables is specified using a graphical model [8]. The graphical representation has several advantages. Among them are the existence of algorithms for inferring the class label (and in general to complete missing data), the ability to intuitively represent fusion of different modalities with the graph structure [9, 10], the ability to perform classification and learning without complete data, and most importantly, the ability to learn with both labeled and unlabeled data. We discuss possible strategies for choosing a good graph structure and argue that in many problems, it is necessary to search for such a structure. Most structure search algorithms are driven by likelihood based cost functions, which are potentially inadequate for classification [11, 12] due to their attempt to maximize the overall likelihood of the data, while largely ignoring the important quantity for classification; the class a-posteriori likelihood. As such, we propose a classification driven stochastic structure search algorithm (SSS), which combines both labeled and unlabeled data to train the classifier and search for a better performing Bayesian network structure.

Following the new understanding of the limitations imposed by the properties of unlabeled data, and equipped with an algorithm to overcome these limitations, we apply the Bayesian network classifiers to to two human-computer interaction problems: facial expression recognition and face detection. In both of these applications, obtaining unlabeled training data is relatively easy; any image can be classified as being a face or not being a face, and any video of humans can contains the appearance of facial expressions. However, in both cases, labeling of the data is difficult. For facial expression recognition, accurate labeling requires expert knowledge [13] and for both applications, labeling of a large amount of data is time consuming for the human labeler. We show that the structure search is beneficial even for relatively small labeled data sets, with large amount of unlabeled data in both of these problems.

The rest of the paper is organized as follows. In Section II we overview learning with labeled and unlabeled data, discuss the value of unlabeled data and illustrate the possibility of unlabeled data to degrade the classification performance. In Section III we propose possible solutions for Bayesian network classifiers to utilize unlabeled data positively by learning the network structure. We introduce a new stochastic structure search algorithm driven by classification performance and empirically show its ability to learn with both labeled and unlabeled data using datasets from the UCI machine learning repository [14]. In Section IV-A we describe the components of our real-time face recognition system, including the real-time face tracking system and the features extracted for classification of facial expressions. We perform experiments of our facial expression recognition system using two databases and show the ability to utilize unlabeled data to enhance the classification performance, even with a small labeled training set. Experiments of Bayesian network classifiers for face detection are given in Section IV-B. We have concluding remarks in Section V.

II. Learning a classifier from labeled and unlabeled training data

This section presents notation, terminology, and describes a mathematical analysis of semi-supervised learning that is used throughout this paper.

The goal is to classify an incoming vector of observables $\mathbf{X}$. Each instantiation of $\mathbf{X}$ is a *sample*. There exists a *class variable* C; the values of C are the *classes*. We want to build *classifiers* that receive a sample $\mathbf{x}$ and output a class. We assume 0-1 loss, and consequently our objective is to minimize the probability of error (*classification error*). If we knew exactly the joint distribution $p(C, \mathbf{X})$, the optimal rule would be to choose the class value with the maximum a-posteriori probability, $p(C|\mathbf{x})$ [15]. This classification rule attains the minimum possible classification error, called the *Bayes error*.

We take that the probabilities of $(C, \mathbf{X})$, or functions of these probabilities, are estimated from data and then "plugged" into the optimal classification rule. We assume that a parametric model $p(C, \mathbf{X}|\theta)$ is adopted. An estimate of θ is denoted by $\hat{\theta}$. If the distribution $p(C, \mathbf{X})$ belongs to the family $p(C, \mathbf{X}|\theta)$, we say the "model is correct"; otherwise we say the "model is incorrect". For Bayesian networks, we say that the assumed structure (the directed acyclic graph) S, is *correct* when it is possible to find a distribution satisfying the Markov condition on S and that matches the distribution that generates data; otherwise, the structure is *incorrect*. We use "estimation bias" loosely to mean the expected difference between $p(C, \mathbf{X})$ and the estimated $p\left(C, \mathbf{X}|\hat{\theta}\right)$.

We consider the following scenario. A sample $(c, \mathbf{x})$ is generated from $p(C, \mathbf{X})$. The value c is then either revealed, and the sample is a *labeled* one; or the value c is hidden, and the sample is an *unlabeled* one. The probability that any sample is labeled, denoted by λ, is fixed, known, and independent of the samples[1]. Thus the same underlying distribution $p(C, \mathbf{X})$ models both labeled and unlabeled data. Given a set of N_l labeled samples and N_u unlabeled samples, we use maximum likelihood for estimating $\hat{\theta}$. We consider distributions that decompose $p(C, \mathbf{X}|\theta)$ as $p(C|\mathbf{X}, \theta)\, p(\mathbf{X}|\theta)$, where both $p(\mathbf{X}|\mathbf{C}, \theta)$ and $p(C|\theta)$ depend explicitly on θ. This is referred to as a *generative model*. The log-likelihood function of a generative model for a dataset with labeled and unlabeled data is:

$$L(\theta) = L_l(\theta) + L_u(\theta) + \log\left(\lambda^{N_l}(1-\lambda)^{N_u}\right), \tag{1}$$

where $L_u(\theta) = \sum_{j=(N_l+1)}^{N_l+N_u} \log\left[p(\mathbf{x_j}|\theta)\right]$, and $L_l(\theta) = \sum_{i=1}^{N_l} \log\left[\prod_C (p(C=c'|\theta)\, p(\mathbf{x_i}|\mathbf{c}', \theta))^{I_{\{C=c'\}}(c_i)}\right]$ with $I_A(Z)$ the indicator function: 1 if $Z \in A$; 0 otherwise. $L_l(\theta)$ and $L_u(\theta)$ are the likelihoods of the labeled and unlabeled data, respectively.

Statistical intuition suggests that it is reasonable to expect an average improvement in classification per-

[1] This is different from [3] and [16], where λ is a parameter that can be set.

formance for any increase in the number of samples (labeled or unlabeled). Indeed, previous theoretical works [17, 18] showed that unlabeled data are always asymptotically useful for classification. Information theoretic arguments provided by [2] and [1] further strengthen the asymptotic arguments. However, in all such works there is an assumption that the model is correct. We performed extensive experiments providing empirical evidence that degradation of performance can occur and is directly related to incorrect modeling assumptions (see [19–21] for a detailed description). To provide a theoretical explanation to the empirical evidence, we derived the asymptotic properties of maximum likelihood estimators for the labeled-unlabeled case. The analysis, presented in the rest of this section, provides a unified explanation of the behavior of classifiers for both cases; when the model is correct and when it is not.

We base our result on the work of White [22] on the properties of maximum likelihood estimators without assuming model correctness. White [22] showed that under suitable regularity conditions, maximum likelihood estimators converge to a parameter set θ^* that minimizes the KL distance between the assumed family of distributions, $p(Y|\theta)$, and the true distribution, $p(Y)$. He also shows that the estimator is asymptotically Normal, i.e., $\sqrt{N}(\hat{\theta}_N - \theta^*) \sim \mathcal{N}(0, C_Y(\theta))$ as N (the number of samples) goes to infinity. $C_Y(\theta)$ is a covariance matrix equal to $A_Y(\theta)^{-1} B_Y(\theta) A_Y(\theta)^{-1}$, evaluated at θ^*, where $A_Y(\theta)$ and $B_Y(\theta)$ are matrices whose (i, j)'th element ($i, j = 1, ...d$, where d is the number of parameters) is given by:

$$\begin{aligned} A_Y(\theta) &= E\left[\partial^2 \log p(Y|\theta) \,/ \partial\theta_i\theta_j\right], \\ B_Y(\theta) &= E[(\partial \log p(Y|\theta) \,/ \partial\theta_i)(\partial \log p(Y|\theta) \,/ \partial\theta_j)] \,. \end{aligned}$$

In semi-supervised learning, the samples are realizations of

$$\begin{cases} (C, \mathbf{X}) & \text{with probability } \lambda; \\ \mathbf{X} & \text{with probability } (1-\lambda). \end{cases} \tag{2}$$

For our analysis, it is convenient to obtain a single expression for both situations. Denote by $\tilde{C}$ a random variable that assumes the same values of C plus the "unlabeled" value u. We have $p\left(\tilde{C} \neq u\right) = \lambda$. The actually observed samples are realizations of $(\tilde{C}, \mathbf{X})$, so we can summarize Expression (2) compactly as follows:

$$\tilde{p}\left(\tilde{C} = c, \mathbf{X} = \mathbf{x}\right) = (\lambda p(C = c, \mathbf{X} = \mathbf{x}))^{I_{\{\tilde{C} \neq u\}}(c)} \left((1-\lambda) p(\mathbf{X} = \mathbf{x})\right)^{I_{\{\tilde{C} = u\}}(c)}, \tag{3}$$

where $p(\mathbf{X})$ is a mixture density obtained from $p(C, \mathbf{X})$. Accordingly, the parametric model adopted for $(\tilde{C}, \mathbf{X})$ is:

$$\tilde{p}\Big(\tilde{C} = c, \mathbf{X} = \mathbf{x}|\theta\Big) = (\lambda p(C = c, \mathbf{X} = \mathbf{x}|\theta))^{I_{\{\tilde{C} \neq u\}}(c)} ((1-\lambda)p(\mathbf{X} = \mathbf{x}|\theta))^{I_{\{\tilde{C}=u\}}(c)} . \tag{4}$$

Using these definitions, we obtain:

Theorem 1: Consider supervised learning where samples are randomly labeled with probability λ. Adopt the regularity conditions in Theorems 3.1, 3.2, 3.3 from [22], with Y replaced by $(C, \mathbf{X})$ and by $\mathbf{X}$, and also assume identifiability for the marginal distributions of $\mathbf{X}$. Then the value of θ^*, the limiting value of maximum likelihood estimates, is:

$$\arg\max_{\theta} \left(\lambda E[\log p(C, \mathbf{X}|\theta)] + (1-\lambda)E[\log p(\mathbf{X}|\theta)]\right), \tag{5}$$

where the expectations are with respect to $p(C, \mathbf{X})$. Additionally, $\sqrt{N}(\hat{\theta}_N - \theta^*) \sim \mathcal{N}(0, C_\lambda(\theta))$ as $N \to \infty$, where $C_\lambda(\theta)$ is given by:

$$\begin{aligned} C_\lambda(\theta) &= A_\lambda(\theta)^{-1} B_\lambda(\theta) A_\lambda(\theta)^{-1} \text{ with,} \qquad (6)\\ A_\lambda(\theta) &= \left(\lambda A_{(C,\mathbf{X})}(\theta) + (1-\lambda)A_{\mathbf{X}}(\theta)\right) \text{ and} \\ B_\lambda(\theta) &= \left(\lambda B_{(C,\mathbf{X})}(\theta) + (1-\lambda)B_{\mathbf{X}}(\theta)\right), \end{aligned}$$

evaluated at θ^*. □

Proof. Theorems 3.1, 3.2, 3.3 from [22], θ^* maximizes $E\Big[\log \tilde{p}\Big(\tilde{C}, \mathbf{X}|\theta\Big)\Big]$ (expectation with respect to $\tilde{p}\Big(\tilde{C}, \mathbf{X}\Big)$). We have:

$$\begin{aligned} E\Big[\log \tilde{p}\Big(\tilde{C}, \mathbf{X}|\theta\Big)\Big] &= E\Big[I_{\{\tilde{C} \neq u\}}(\tilde{C}) \left(\log \lambda + \log p(C, \mathbf{X}|\theta)\right) + I_{\{\tilde{C}=u\}}(\tilde{C}) \left(\log(1-\lambda) + \log p(\mathbf{X}|\theta)\right)\Big] \\ &= \lambda \log \lambda + (1-\lambda)\log(1-\lambda) + \\ &\quad E\Big[I_{\{\tilde{C} \neq u\}}(\tilde{C}) \log p(C, \mathbf{X}|\theta)\Big] + E\Big[I_{\{\tilde{C}=u\}}(\tilde{C}) \log p(\mathbf{X}|\theta)\Big] . \end{aligned}$$

The first two terms of this expression are irrelevant to maximization with respect to θ. The last two terms are equal to

$$\lambda E\Big[\log p(C, \mathbf{X}|\theta) \,|\tilde{C} \neq u\Big] + (1-\lambda)E\Big[\log p(\mathbf{X}|\theta) \,|\tilde{C} = u\Big] .$$

As we have $\tilde{p}\Big(\tilde{C}, \mathbf{X} | \tilde{C} \neq u\Big) = p(C, \mathbf{X})$ and $\tilde{p}\Big(\mathbf{X} | \tilde{C} = u\Big) = p(\mathbf{X})$ (Expression (3)), the last expression is equal to

$$\lambda E[\log p(C, \mathbf{X}|\theta)] + (1 - \lambda) E[\log p(\mathbf{X}|\theta)]\,,$$

where the last two expectations are now with respect to $p(C, \mathbf{X})$. Thus we obtain Expression (5). Expression (6) follows directly from White's theorem and Expression (5), replacing Y by $(C, \mathbf{X})$ and $\mathbf{X}$ where appropriate.□

Expression (5) indicates that semi-supervised learning can be viewed asymptotically as a "convex" combination of supervised and unsupervised learning. The objective function for semi-supervised learning is a combination of the objective function for supervised learning ($E[\log p(C, \mathbf{X}|\theta)]$) and the objective function for unsupervised learning ($E[\log p(\mathbf{X}|\theta)]$).

Denote by θ^*_λ the value of θ that maximizes Expression (5) for a given λ. Then θ^*_1 is the asymptotic estimate of θ for *supervised* learning, denoted by θ^*_l. Likewise, θ^*_0 is the asymptotic estimate of θ for *unsupervised* learning, denoted by θ^*_u.

The asymptotic covariance matrix is positive definite as $B_Y(\theta)$ is positive definite and $A_Y(\theta)$ is symmetric for any Y, $\theta A(\theta)^{-1} B_Y(\theta) A(\theta)^{-1} \theta^T = w(\theta) B_Y(\theta) w(\theta)^T > 0$, where $w(\theta) = \theta A_Y(\theta)^{-1}$. We see that asymptotically, an increase in N, the number of labeled and unlabeled samples, will lead to a reduction in the variance of $\hat{\theta}$.

Such a guarantee can perhaps be the basis for the optimistic view that unlabeled data should always be used to improve classification accuracy. In the following, we show this view is valid when the model is correct, and that it is not always valid when the model is incorrect.

A. Model is correct

Suppose first that the family of distributions $P(C, \mathbf{X}|\theta)$ contains the distribution $P(C, \mathbf{X})$; that is, $P(C, \mathbf{X}|\theta_\top) = P(C, \mathbf{X})$ for some $\theta_\top$. Under this condition, the maximum likelihood estimator is consistent, thus, $\theta^*_l = \theta^*_u = \theta_\top$ given identifiability. Thus, $\theta^*_\lambda = \theta_\top$ for any $0 \leq \lambda \leq 1$.

Consider the Taylor expansion of the classification error around $\theta_\top$, as suggested by Shahshahani and Landgrebe [1], linking the decrease in variance associated with unlabeled data to a decrease in classification error, and assuming existence of necessary derivatives:

$$\mathbf{e}(\hat{\theta}) \approx \mathbf{e}_{\mathrm{B}} + \left.\frac{\partial \mathbf{e}(\theta)}{\partial \theta}\right|_{\theta_\top} \left(\hat{\theta} - \theta_\top\right) + \frac{1}{2}\mathrm{tr}\left(\left.\frac{\partial^2 \mathbf{e}(\theta)}{\partial \theta^2}\right|_{\theta_\top} \left(\hat{\theta} - \theta_\top\right)\left(\hat{\theta} - \theta_\top\right)^T \right). \quad (7)$$

Take expected values on both sides. Asymptotically the expected value of the second term in the expansion is zero, as maximum likelihood estimators are asymptotically unbiased when the model is correct. Shahshahani and Landgrebe thus argue that

$$E\left[\mathbf{e}(\hat{\theta})\right] \approx \mathbf{e}_{\mathrm{B}} + (1/2)\mathrm{tr}\left((\partial^2 \mathbf{e}(\theta)/\partial \theta^2)|_{\theta_\top} \mathrm{Cov}(\hat{\theta}) \right)$$

where $\mathrm{Cov}(\hat{\theta}_N)$ is the covariance matrix for $\hat{\theta}_N$ and $\mathbf{e}_{\mathrm{B}} = \mathbf{e}(\theta_\top)$ is the Bayes error rate. They show that if $\mathrm{Cov}(\theta') \geq \mathrm{Cov}(\theta'')$ for some θ', θ'', then the second term in the approximation is larger for θ' than for θ''. Because $\mathbf{I}_u(\theta)$ is always positive definite, $\mathbf{I}_l(\theta) \leq \mathbf{I}(\theta)$. Thus, using the Cramer-Rao lower bound,

$$\mathrm{Cov}(\hat{\theta}_N) \geq \frac{1}{N}(\mathbf{I}(\theta))^{-1},$$

the covariance with labeled and unlabeled data is smaller than the covariance with just labeled data, leading to the conclusion that *unlabeled data must cause a reduction in classification error when the model is correct*. It should be noted that this argument holds as the number of records goes to infinity, and is an approximation for finite values.

A more formal, but less general, argument is presented by Ganesalingam and McLachlan [23] as they compare the relative efficiency of labeled and unlabeled data. Castelli [17] also derives a Taylor expansion of the classification error, to study estimation of the mixing factors, $p(C = c)$; the derivation is very precise and states all the required assumptions.

B. Model is incorrect

We now study the more realistic scenario where the distribution $P(C, \mathbf{X})$ does not belong to the family of distributions $P(C, \mathbf{X}|\theta)$. In view of Theorem 1, it is perhaps not surprising that unlabeled data can have the deleterious effect observed occasionally in the literature. Suppose that $\theta_u^* \neq \theta_l^*$ and that $\mathbf{e}(\theta_u^*) > \mathbf{e}(\theta_l^*)$, as in the example in the next section.[2] If we observe a large number of labeled samples, the classification error

[2] We have to handle a difficulty with $\mathbf{e}(\theta_u^*)$: given only unlabeled data, there is no information to decide the labels for decision regions, and then the classification error is 1/2 [17]. Instead of actually using $\mathbf{e}(\theta_u^*)$, we could consider $\mathbf{e}(\theta_\epsilon^*)$ for any value of $\epsilon > 0$. To simplify the discussion, we avoid the complexities of $\mathbf{e}(\theta_\epsilon^*)$ by assuming that, when $\lambda = 0$, an "oracle" will be available to indicate the labels of the decision regions.

is approximately $\mathbf{e}(\theta_l^*)$. If we then collect more samples, most of which unlabeled, we eventually reach a point where the classification error approaches $\mathbf{e}(\theta_u^*)$. So, the net result is that we started with classification error close to $\mathbf{e}(\theta_l^*)$, and by adding a large number of unlabeled samples, classification performance degraded. The basic fact here is that estimation and classification bias are affected differently by different values of λ. Hence, a necessary condition for this kind of performance degradation is that $\mathbf{e}(\theta_u^*) \neq \mathbf{e}(\theta_l^*)$; a sufficient condition is that $\mathbf{e}(\theta_u^*) > \mathbf{e}(\theta_l^*)$.

The focus on asymptotics is adequate as we want to eliminate phenomena that can vary from dataset to dataset. If $\mathbf{e}(\theta_l^*)$ is smaller than $\mathbf{e}(\theta_u^*)$, then a large enough labeled dataset can be dwarfed by a much larger unlabeled dataset — the classification error using the whole dataset can be larger than the classification error using the labeled data only.

B.1 Example: Bivariate Gaussians with spurious correlation

The previous discussion alluded to the possibility that $\mathbf{e}(\theta_u^*) > \mathbf{e}(\theta_l^*)$ when the model is incorrect. To the skeptical reader who still may think that this won't occur in practice, or that numerical algorithms, such as EM, are to blame, we analytically show how this occurs with an example of obvious practical significance. More examples are provided in [21] and [20].

We will assume that bivariate Gaussian samples (X, Y) are observed. The only modeling error is an ignored dependency between observables. This type of modeling error is quite common in practice and has been studied in the context of supervised learning [24, 25]. Is it often argued that ignoring some dependencies can be a positive decision, as we may see a reduction in the number of parameters to be estimated and a reduction on the variance of estimates [26].

Example 1: Consider real-valued observations (X, Y) taken from two classes c' and c''. We know that X and Y are Gaussian variables, and we know their means and variances given the class C. The mean of (X, Y) is $(0, 3/2)$ conditional on $\{C = c'\}$, and $(3/2, 0)$ conditional on $\{C = c''\}$. Variances for X and for Y conditional on C are equal to 1. We do not know, and have to estimate, the mixing factor $\eta = p(C = c')$. The data is sampled from a distribution with mixing factor equal to 3/5.

We want to obtain a Naive-Bayes classifier that can approximate $p(C|X, Y)$; Naive-Bayes classifiers are based on the assumption that X and Y are independent given C. Suppose that X and Y are independent conditional on $\{C = c'\}$ but that X and Y are dependent conditional on $\{C = c''\}$. This dependency is manifested by a correlation $\rho = E[(X - E[X])(Y - E[Y])] = 4/5$. If we knew the value of ρ, we would obtain an optimal classification boundary on the plane $X \times Y$. This optimal classification boundary is shown in Figure 1, and is defined by the function

$$y = \left(40x - 87 + \sqrt{5265 - 2160x + 576x^2 + 576\log(100/81)}\right)/32.$$

Under the incorrect assumption that $\rho = 0$, the classification boundary is then linear:

$$y = x + 2\log((1 - \hat{\eta})/\hat{\eta})/3,$$

and consequently it is a decreasing function of $\hat{\eta}$. With labeled data we can easily obtain $\hat{\eta}$ (a sequence of Bernoulli trials); then $\eta_l^* = 3/5$ and the classification boundary is given by $y = x - 0.27031$.

Note that the (linear) boundary obtained with labeled data is not the best possible linear boundary. We can in fact find the best possible linear boundary of the form $y = x + \gamma$. For any γ, the classification error $\mathbf{e}(\gamma)$ is

$$\frac{3}{5}\int_{-\infty}^{\infty}\int_{-\infty}^{x+\gamma} N\left(\begin{bmatrix} 0 \\ 3/2 \end{bmatrix}, \text{diag}[1,1]\right) dydx + \frac{2}{5}\int_{-\infty}^{\infty}\int_{x+\gamma}^{\infty} N\left(\begin{bmatrix} 3/2 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 4/5 \\ 4/5 & 1 \end{bmatrix}\right) dydx.$$

By interchanging differentiation with respect to γ with integration, it is possible to obtain $d\mathbf{e}(\gamma)/d\gamma$ in closed form. The second derivative $d^2\mathbf{e}(\gamma)/d\gamma^2$ is positive when $\gamma \in [-3/2, 3/2]$; consequently there is a single minimum that can be found by solving $d\mathbf{e}(\gamma)/d\gamma = 0$. We find the minimizing γ to be $(-9 + 2\sqrt{45/4 + \log(400/81)})/4 \approx -0.45786$. The line $y = x - 0.45786$ is the best linear boundary for this problem. If we consider the set of lines of the form $y = x + \gamma$, we see that the farther we go from the best line, the larger the classification error. Figure 1 shows the linear boundary obtained with labeled data and the best possible linear boundary. The boundary from labeled data is "above" the best linear boundary.

Now consider the computation of η_u^*, the asymptotic estimate with unlabeled data:

$$\begin{aligned}\eta_u^* \;=\; & \arg\max_{\eta\in[0,1]} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \log\left(\eta N([0,3/2]^T, \text{diag}[1,1]) + (1-\eta)N([3/2,0]^T, \text{diag}[1,1])\right) \\ & \left((3/5)N([0,3/2]^T, \text{diag}[1,1]) + (2/5)N\left(\begin{bmatrix} 3/2 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 4/5 \\ 4/5 & 1 \end{bmatrix}\right)\right) dydx.\end{aligned}$$

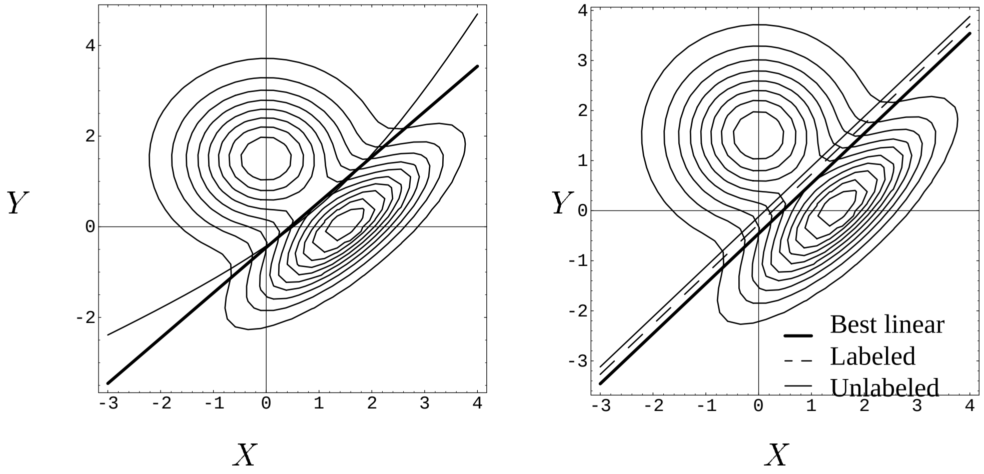


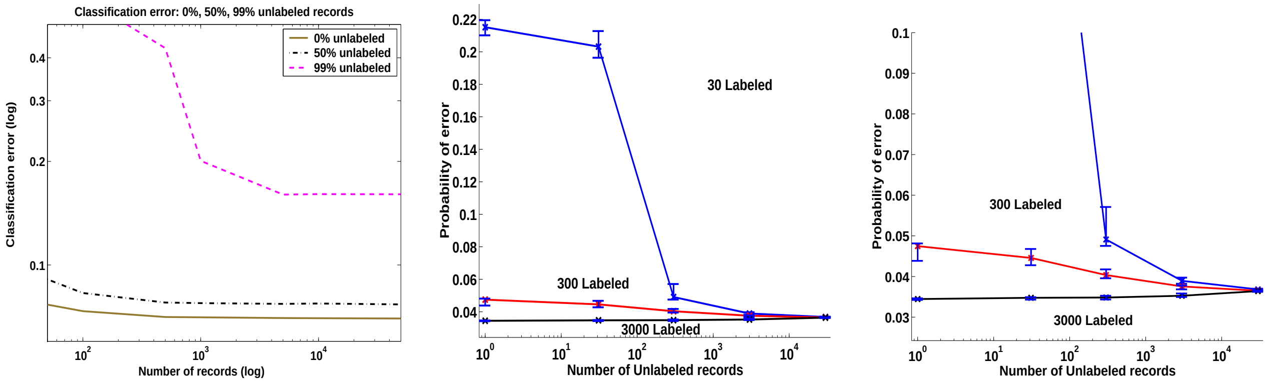
Fig. 1. Graphs for Example 1. On the left, contour plots of the mixture $p(X, Y)$, the optimal classification boundary (quadratic curve) and the best possible classification boundary of the form $y = x + \gamma$. On the right, the same contour plots, and the best linear boundary (lower line), the linear boundary obtained from labeled data (middle line) and the linear boundary obtained from unlabeled data (upper line); thus the classification error of the unlabeled classifier is larger than that of the labeled classifier

The second derivative of this double integral is always negative (as can be seen interchanging differentiation with integration), so the function is concave and there is a single maximum. We can search for the zero of the derivative of the double integral with respect to η. We obtain this value numerically, $\eta^*_u \approx 0.54495$. Using this estimate, the linear boundary from unlabeled data is $y = x - 0.12019$. This line is "above" the linear boundary from labeled data, and, given the previous discussion, leads to a larger classification error than the boundary from unlabeled data. We have: $\mathbf{e}(\gamma) = 0.06975$; $\mathbf{e}(\theta^*_l) = 0.07356$; $\mathbf{e}(\theta^*_u) = 0.08141$. The boundary obtained from unlabeled data is also shown in Figure 1. □

This example suggests the following situation. Suppose we collect a large number N_l of labeled samples from $p(C, X)$, with $\eta = 3/5$ and $\rho = 4/5$. The labeled estimates form a sequence of Bernoulli trials with probability $3/5$, so the estimates quickly approach η^*_l (the variance of $\hat{\eta}$ decreases as $6/(25N_l)$). If we add a very large amount of unlabeled data to our data, $\hat{\eta}$ approaches η^*_u and the classification error increases.

C. Finite sample effects

The asymptotic analysis of semi-supervised learning suffices to show the fundamental problem that can occur when learning with unlabeled data. But what occurs with finite sample size datasets? We performed extensive experiments with real and artificial datasets of various sizes, described in [7, 20]. Here we bring



(a) (b) (c)

Fig. 2. (a) LU-graphs for the example with two Gaussian observables. Each sample in each graph is the average of 100 trials; classification error was obtained by testing in 10000 labeled samples drawn from the correct model. (b) Naive Bayes classifiers from data generated from a TAN model (introduced in Section III) with 49 observables (each variable with 2 to 4 values); points in the graphs summarizes 10 runs on testing data (bars cover 30 to 70 percentiles). (c) Same graph as (b), enlarged. Note that unlabeled data does lead to a significant improvement in performance when added to 30 or 300 labeled samples. There is performance degradation in the presence of 3000 labeled samples.

some of the main results. Throughout our experiments we used the EM algorithm [27] to maximize the likelihood (Expression (1)) [27], and we start the EM algorithm with the parameters obtained using labeled data, as these starting points can be obtained in closed-form.

To visualize the effect of labeled and unlabeled samples, we suggest that the most profitable strategy is to fix the *percentage* of unlabeled samples (λ) among all training samples. We then plot classification error against the number of training samples. Call such a graph a *LU-graph*.

Example 2: Consider a situation where we have a binary class variable C with values c' and c'', and $p(C = c') = 0.4017$. We also have two real-valued observables X and Y with distributions:

$$p(X|c') = N(2,1), \quad p(X|c'') = N(3,1),$$

$$p(Y|c',x) = N(2,1), \quad p(Y|c'',x) = N(1+2x,1).$$

There is dependency between Y and X conditional on $\{C = c''\}$. Suppose we build a Naive Bayes classifier for this problem. Figure II-C(a) shows the LU-graphs for 0% unlabeled samples, 50% unlabeled samples and 99% unlabeled samples, averaging over a large ensemble of classifiers. As expected, the asymptotes converge to different values. Suppose then that we started with 50 labeled samples as our training data. Our classification error would be about 7.8%, as we can see in the LU-graph for 0% unlabeled data. Suppose we added 50 labeled samples; we would obtain a classification error of about 7.2%. Now suppose we added 100 *unlabeled* samples. We would move from the 0% LU-graph to the 50% LU-graph. Classification error would increase to 8.2%! And if we then added 9800 unlabeled samples, we would move to the 99% LU-graph, with classification error about 16.5% — more than twice the error we had with just 50 labeled samples. □

It should be noted that in difficult classification problems, where LU-graphs decrease very slowly, unlabeled data may improve classification performance for certain regions of the LU graphs. Problems with a large number of observables and parameters should require more training data, so we can expect that such problems benefit more consistently from unlabeled data. Figures II-C(b-c) illustrate this possibility for Naive-Bayes classifiers with 49 features. Another possible phenomenon is that the addition of a substantial number of unlabeled samples may reduce variance and decrease classification error, but an additional, much larger, pool of unlabeled data can eventually add enough bias so as to increase classification error. Such a situation is likely to have happened in some of the results reported by Nigam et al [3], where classification errors go up and down as more unlabeled samples are added.

D. Short summary

To summarize the results so far, we can say the following:

- Labeled and unlabeled data contribute to a reduction in variance in semi-supervised learning under maximum likelihood estimation.
- When the model is correct, the maximum likelihood estimator is unbiased and both labeled and unlabeled data contribute to a reduction in classification error by reducing variance. Also, unlabeled data suffice to define the decision regions and labeled data can be used solely to label the regions.
- When the model is incorrect, there may be different asymptotic estimation biases for different values of λ. Asymptotic classification error may also be different for different values of λ. An increase in the number of unlabeled samples may lead to a larger estimation bias and a larger classification error. Example 1 illustrated this possibility.
- For finite size datasets and incorrect models, unlabeled data can be observed to improve or degrade the classification performance, a behavior that can be characterized by the LU-graphs.

In essence, semi-supervised learning displays an odd failure of robustness: for certain modeling errors, more unlabeled data can degrade classification performance. Estimation bias is the central factor in this phenomenon, as the level of bias depends on the ratio of labeled to unlabeled samples. Most existing theoretical results on semi-supervised learning are based on the assumption of no modeling error, and consequently bias has not been an issue so far.

III. Semi-supervised learning for Bayesian network classifiers

We now turn our attention to the implication of the analysis to Bayesian network classifiers. As stated before, we chose Bayesian network classifiers for several reasons; classification is possible with missing data in general and unlabeled data in particular, the graphical representation is intuitive and can be easily expanded to add different features and modalities for some applications, and there are efficient algorithms for inference. Other popular classifiers, such as support vector machines and Neural networks, cannot currently handle unlabeled data, and therefore we do not consider them in this paper.

The conclusion of the previous section indicates the importance of obtaining the correct model when using unlabeled data to learn a classifier. In the context of Bayesian networks, finding the correct model amounts to obtaining a correct structure. If a correct structure is obtained, unlabeled data improve a classifier; otherwise, unlabeled data can actually degrade performance. Somewhat surprisingly, the option of searching for better structures has not been proposed by researchers that have previously witnessed the performance degradation when learning with unlabeled data.

Bayesian networks [8] have become popular in recent years as tools for modeling and classification. A Bayesian network is composed of a directed acyclic graph in which every node is associated with a variable X_i and with a conditional distribution $p(X_i|\Pi_i)$, where Π_i denotes the parents of X_i in the graph. The joint probability distribution is factored to the collection of conditional probability distributions of each node in the graph as:

$$p(X_1, ..., X_n) = \prod_{i=1}^{n} p(X_i|\Pi_i) \, .$$

The directed acyclic graph is the *structure*, and the distributions $p(X_i|\Pi_i)$ represent the *parameters* of the network. We say that the assumed structure for a network, S', is *correct* when it is possible to find a distribution,

$p(C, \mathbf{X}|S')$, that matches the distribution that generates data, $p(C, \mathbf{X})$; otherwise, the structure is *incorrect*[3],[4]. Maximum likelihood estimation is one of the main methods to learn the parameters of the network. When there are missing data in training set, the Expectation Maximization (EM) algorithm [27] can be used to maximize the likelihood.

As a direct consequence of the analysis in the previous section, a Bayesian network that has the correct structure and the correct parameters is also optimal for classification because the a-posteriori distribution of the class variable is accurately represented.

A Bayesian network classifier is *generative* when the class variable is an ancestor (e.g., parent) of some or all features. A Bayesian network classifier is *diagnostic*, when the class variable has non of the features as descendants. As we are interested in using unlabeled data in learning the Bayesian network classifier, we restrict ourselves to generative structures, and exclude structures that are diagnostic.

A. Switching between generative models: Naive Bayes and TAN

One attempt to overcome the performance degradation from unlabeled data could be to switch models as soon as degradation is detected. Suppose then that we learn a classifier with labeled data only, and we observe a degradation in performance when the classifier is learned with labeled and unlabeled data. We can switch to a more complex structure at that point. As we saw in the previous chapter, bias and variance play an important role in the utilization of unlabeled data. To preserve the balance between the bias from the true distribution and the variance we might want to use a small subset of simple models which can be learned efficiently.

We start with the simplest generative structure, the Naive Bayes. Despite having a non-zero classification bias, the Naive-Bayes classifier performs well for many cases, *when trained with labeled data*. The success is explained in the literature using several arguments; e.g., trade-offs between classification bias and variance when learning with scarce data [26] and tendency of many distributions to be close (in the Kullback-Leibler sense) to the product distribution of the Naive Bayes classifier [28]. However, in semi-supervised learning, the same success is not always observed (see experiments).

If a problem is such that Naive Bayes classifiers suffer from performance degradation with unlabeled data,

[3] These definitions follow directly from the definitions of correct and incorrect models described in the previous section.

[4] There isn't necessarily a unique correct structure, e.g., if a structure is correct (as defined above), all structures that are from the same Markov equivalent class are also correct since causality is not an issue.

we should then switch to a larger family of models. The most promising such family is represented by TAN classifiers, in which the class variable is a parent of all of the observables, and the observables are connected so as to form a tree. Friedman et al. [11] showed that learning the most likely TAN structure can be done efficiently using the Chow-Liu algorithm [29]. An important extension of the TAN learning algorithm was proposed by Meila [30], who constructed an EM algorithm (which we call EM-TAN [7]) for learning with both labeled and unlabeled data. The algorithm enjoys the efficiency of the supervised TAN algorithm, while guaranteeing convergence to a local maximum of the likelihood function.

We have observed that EM-TAN produces classifiers that in practice regularly surpass Naive Bayes classifiers. Still, performance degradation can still occur both for Naive Bayes and TAN (as actually observed in Table I). In such cases, we are faced with several options. The first is to discard the unlabeled data and use only the available labeled samples. The other options are discussed in the next sections.

B. Beyond Naive Bayes and TAN classifiers: unrestricted structure learning

If we observe performance degradation, we may try to find the "correct" structure for our Bayesian network classifier — if we do so, we can profitably use unlabeled data. Alas, learning Bayesian network structure is not a trivial task. We begin by investigating the behavior of structure learning algorithms in the context of semi-supervised learning, presenting new algorithms where needed, and deriving new techniques that improve on existing methods.

B.1 Independence-based methods

The first class of structure learning methods we consider is the class of independence-based methods, also known as constraint-based or test-based methods. There are several such algorithms; a relevant subset is composed of the PC algorithm [31], the IC algorithm [32], and the CBL1 and CBL2 algorithms [33]. All of them can obtain the correct structure if there are fully reliable independence tests available; however not all of them are appropriate for classification. For example, the PC algorithm starts with a fully connected network, and has the tendency to generate structures that are "too dense" (consequently requiring many parameters to be learned, negatively affecting the variance of estimated quantities and increasing the classification error).

The CBL1 and CBL2 algorithms seem particularly well-suited for classification, as they strive to keep the number of edges in the Bayesian networks as small as possible. The performance of CBL1 on labeled data only has been reported to surpass the performance of TAN, even with arbitrary node orderings [34]. Conceptually CBL1 and CBL2 are similar, with CBL1 requiring an ordering to start. We used conditional independence (CI) tests based on mutual information: we declare variables X and Y to be independent given variable Z when their mutual information conditional on Z is smaller than a constant ϵ, which we set to 0.01.

A few modifications are necessary to adapt CBL1 and CBL2 for semi-supervised learning. First, the algorithms are started with a Naive Bayes classifier, and in CBL1 arcs from the class variable to observed variables are allowed to be removed, leading to some restricted forms of feature selection. More importantly, a simple method to generate orderings for CBL1 is developed by generating a fixed number of random orderings, and running the algorithm for all of them. Because CBL1 is quite fast, hundreds of candidate orderings are easily tested, selecting the one that produces the best classifier (using either testing data or cross-validation to select the classifier, depending on the amount of available labeled data).

Because independence-based algorithms like CBL1 do not explicitly optimize a metric, they cannot handle unlabeled data directly through an optimization scheme like EM. To handle unlabeled data, the following strategy was opted (denoted as EM-CBL): Start by learning a Bayesian network with the available labeled data; then use EM to process unlabeled data followed by independence tests with the "probabilistic labels" generated by EM, to obtain a new structure. EM is used again in the new structure and the cycle is repeated, until two subsequent networks are identical. It should be noted that such a scheme, however intuitively reasonable, has no convergence guarantees; one test even displayed oscillating behavior.

Despite such difficulties, EM-CBL1 has been observed to actually improve the performance obtained with EM-TAN in many problems (see experiments). This apparent victory must be taken carefully though: the algorithm takes much more computational effort than EM-TAN, and its improvement over EM-TAN is only marginal. Moreover, the algorithm relies on the computation of mutual information with the "probabilistic labels" generated by EM; such a method has been observed to lead to unreliable CI tests. Given the fact that all independence-based algorithms depend critically on these tests, the lack of robustness of such tests creates difficulties for EM-CBL1 in several classification problems. The EM-CBL2 has been observed to be consistently *worse* than EM-TAN, hence it was not explored further.

To conclude, experience shows that the use of independence-based methods in semi-supervised learning is not promising.

B.2 Likelihood and Bayesian Score-based methods

Here we turn to a different family of algorithms, those based on scores. At the heart of most score based methods is the likelihood of the training data. To avoid overfitting the model to the data, likelihood is offset by a complexity penalty term, such as the minimum description length (MDL), Bayesian information criterion (BIC) and others. A good comparison of the different methods is found in [35]. Most existing methods cannot, in their present form, handle missing data in general and unlabeled data in particular. The structural EM (SEM) algorithm [36] is one attempt to learn structure with missing data. The algorithm attempts to maximize the Bayesian score using an EM-like scheme in the space of structures and parameters; the method performs an always-increasing search in the space of structures, but does not guarantee the attainment of even a local maximum. Algorithms using other scores could most likely be extended to handle unlabeled data in much the same way as the SEM algorithm.

When learning the structure of a classifier, score based structure learning approaches (such as BIC and MDL) have been strongly criticized. The problem is that with finite amounts of data, the a-posteriori probability of the class variable can have a small effect on the score, that is dominated by the marginal of the observables, therefore leading to poor classifiers [11, 12]. Friedman et al. [11] showed that TAN surpasses score-based methods for the fully labeled case, *when learning classifiers*. The point is that with unlabeled data, score-based methods such as SEM are likely to go astray even more than it has been reported in the supervised case; the marginal of the observables further dominates the likelihood portion of the score as the ratio of unlabeled data increases.

Bayesian approaches to structure learning have also been proposed in [37, 38]. Madigan and York [38] construct a Markov Chain Monte Carlo (MCMC) over the space of possible structures, with the stationary distribution being the posterior of the structures given the data. Metropolis-Hastings sampling [39] is used to sample from the posterior distribution. Friedman and Koller [37] use a two step method in their sampling — first they sample from the distribution over the ordering of the variables followed by exact computation of the desired posterior given the ordering. As with likelihood scores, we can expect these two methods to face difficulties when learning classifiers, since they focus on the joint distribution given the data, and not on the classification error or the a-posteriori probability of the class variable.

C. Classification driven stochastic structure search (SSS)

Both the score-based and independence-based methods try to find the correct structure of the Bayesian network, but fail to do so because there is not enough data for either reliable independence tests or for a search that yields a good classifier. Consider the following alternative. As we are interested in finding a structure that performs well as a classifier, it would be natural to design algorithms that use classification error as the guide for structure learning. Here we can further leverage on the properties of semi-supervised learning: we know that unlabeled data can indicate incorrect structure through degradation of classification performance, and we also know that classification performance improves with the correct structure. Thus, a structure with higher classification accuracy over another indicates an improvement towards finding the optimal classifier.

To learn the structure using classification error, we must adopt a strategy of searching through the space of all structures in an efficient manner while avoiding local maxima. In this section, we propose a method that can effectively search for better structures *with an explicit focus on classification*. We essentially need to find a search strategy that can efficiently search through the space of structures. As we have no simple closed-form expression that relates structure with classification error, it would be difficult to design a gradient descent algorithm or a similar iterative method. Even if we did that, a gradient search algorithm would be likely to find a local minimum because of the size of the search space.

First we define a measure over the space of structures which we want to maximize:

Definition 1: The *inverse error measure* for structure S' is

$$inv_e(S') = \frac{\frac{1}{p_{S'}(\hat{c}(X) \neq C)}}{\sum_S \frac{1}{p_S(\hat{c}(X) \neq C)}}, \tag{8}$$

where the summation is over the space of possible structures and $p_S(\hat{c}(X) \neq C)$ is the probability of error of the best classifier learned with structure S.

We use Metropolis-Hastings sampling [39] to generate samples from the inverse error measure, without having to ever compute it for all possible structures. For constructing the Metropolis-Hastings sampling, we define a neighborhood of a structure as the set of directed acyclic graphs to which we can transit in the next step. Transition is done using a predefined set of possible changes to the structure; at each transition a change consists of a single edge addition, removal or reversal. We define the acceptance probability of a candidate structure, S_{new}, to replace a previous structure, S_t as follows:

$$\min\left(1, \left(\frac{inv_e(S^{new})}{inv_e(S^t)}\right)^{1/T} \frac{q(S^t|S^{new})}{q(S^{new}|S^t)}\right) = \min\left(1, \left(\frac{p^t_{error}}{p^{new}_{error}}\right)^{1/T} \frac{N_t}{N_{new}}\right), \tag{9}$$

where $q(S'|S)$ is the transition probability from S to S' and N_t and N_{new} are the sizes of the neighborhoods of S_t and S_{new} respectively; this choice corresponds to equal probability of transition to each member in the neighborhood of a structure. This choice of neighborhood and transition probability creates a Markov chain which is aperiodic and irreducible, thus satisfying the Markov chain Monte Carlo (MCMC) conditions [38]. We summarize the algorithm, which we name stochastic structure search (SSS), in Figure 3.

```
Procedure Stochastic structure search (SSS):
• Fix the network structure to some initial structure, $S_0$.
• Estimate the parameters of the structure $S_0$ and compute the probability of error $p^0_{error}$.
• Set $t = 0$.
• Repeat, until a maximum number of iterations is reached ($MaxIter$),
  – Sample a new structure $S_{new}$, from the neighborhood of $S_t$ uniformly, with probability $1/N_t$.
  – Learn the parameters of the new structure using maximum likelihood estimation. Compute the probability of error of the new classifier, $p^{new}_{error}$.
  – Accept $S_{new}$ with probability given in Eq.(9).
  – If $S_{new}$ is accepted, set $S_{t+1} = S_{new}$ and $p^{t+1}_{error} = p^{new}_{error}$ and change $T$ according to the temperature decrease schedule. Otherwise $S_{t+1} = S_t$.
  – $t = t + 1$.
• return the structure $S_j$, such that $j = \arg\min_{0 \le j \le MaxIter}(p^j_{error})$.
```

Fig. 3. Stochastic structure search algorithm

We add T as a temperature factor in the acceptance probability. Roughly speaking, T close to 1 would allow acceptance of more structures with higher probability of error than previous structures. T close to 0 mostly allows acceptance of structures that improve probability of error. A fixed T amounts to changing the distribution being sampled by the MCMC, while a decreasing T is a simulated annealing run, aimed at finding the maximum of the inverse error measures. The rate of decrease of the temperature determines the rate of convergence. Asymptotically in the number of data, a logarithmic decrease of T guarantees convergence to a global maximum with probability that tends to one [40].

The SSS algorithm, with a logarithmic cooling schedule T, can find a structure that is close to minimum probability of error. There are two caveats though. First, the logarithmic cooling schedule is very slow. We use faster cooling schedules and a starting point which is the best out of either the NB classifier or the TAN classifier. Second, we never have access to the true probability of error for any given structure, p^S_{error}. Instead, we use the empirical error over the training data (denoted as $\hat{p}^S_{error}$).

To avoid the problem of overfitting several approaches are possible. The first is cross-validation; the labeled training data is split to smaller sets and several tests are performed using the smaller sets as test sets. However, this approach can significantly slow down the search, and is suitable only if the labeled training set is moderately large. Another approach is to penalize different structures according to some complexity measure. We could use the BIC or MDL complexity measure, but we chose to use use the multiplicative penalty term derived from structural risk minimization since it is directly related to the relationship between training error and generalization error. We define a modified error term for use in Eq. (8) and (9):

$$(\hat{p}^S_{error})^{mod} = \frac{\hat{p}^S_{error}}{1 - c \cdot \sqrt{\frac{h_S(log\frac{2n}{h_S}+1)-log(\eta/4)}{n}}}, \tag{10}$$

where h_S is the Vapnik-Chervonenkis (VC) dimension of the classifier with structure S, n is the number of training records, η and c are between 0 and 1.

To approximate the VC dimension, we use $h_S \propto N_S$, where N_S is the number of (free) parameters in the Markov blanket of the class variable in the network, assuming that all variables are discrete. We point the reader to [41], in which it was shown that the VC dimension of a Naive Bayes classifier is linearly proportional to the number of parameters. It is possible to extend this result to networks where the features are all descendants of the class variable. For more general networks, features that are not in the Markov blanket of the class variable cannot effect its value in classification (assuming there are no missing values for any feature), justifying the above approximation. In our initial experiments, we found that the multiplicative penalty outperformed the holdout method and MDL and BIC complexity measures.

D. Evaluation using UCI machine learning datasets

To evaluate structure learning methods with labeled and unlabeled data, we started with an empirical study involving simulated data. We artificially generated data to investigate: (1) whether the SSS algorithm finds a structure that is close to the structure that generated the data, and (2) whether the algorithm uses unlabeled data to improve the classification performance. A typical result is as follows. We generated data from a TAN structure with 10 features. The dataset consisted of 300 labeled and 30000 unlabeled records. We first estimated the Bayes error rate by learning with the correct structure and with a very large fully labeled dataset. We obtained a classification accuracy of 92.49%. We learned one Naive Bayes classifier only with the labeled records, and another with both labeled and unlabeled records; likewise, we learned a TAN classifier only with the labeled records, and another with both labeled and unlabeled records, using the EM-TAN algorithm; and finally, we learned a Bayesian network classifier with our SSS algorithm using both labeled and unlabeled records. The results are presented in the first row of Table I. With the correct structure, adding unlabeled data improves performance significantly (columns TAN-L and EM-TAN). Note that adding unlabeled data degraded the performance from 16% error to 40% error when we learned the Naive Bayes classifier. The structure search algorithm comes close to the performance of the classifier learned with the correct structure. Figure 4(a) shows the changes in the test and train error during the search process. The graph shows the first 600 moves of the search, initialized with the Naive Bayes structure. The error usually decreases as new structures are accepted; occasionally we see an increase in the error allowed by Metropolis-Hastings sampling.

Next, we performed experiments with some of the UCI datasets, using relatively small labeled sets and large unlabeled sets (Table I). The results suggest that structure learning holds the most promise in utilizing the unlabeled data. There is no clear ’winner’ approach, although SSS yields better results in most cases. We see performance degradation with NB for every dataset. EM-TAN can sometimes improve performance over TAN with just labeled data (Shuttle). With the Chess dataset, discarding the unlabeled data and using only TAN seems the best approach. We have compared two likelihood based structure learning methods (K2 and MCMC) on the same datasets as well [20], showing that even if we allow the algorithms to use large labeled

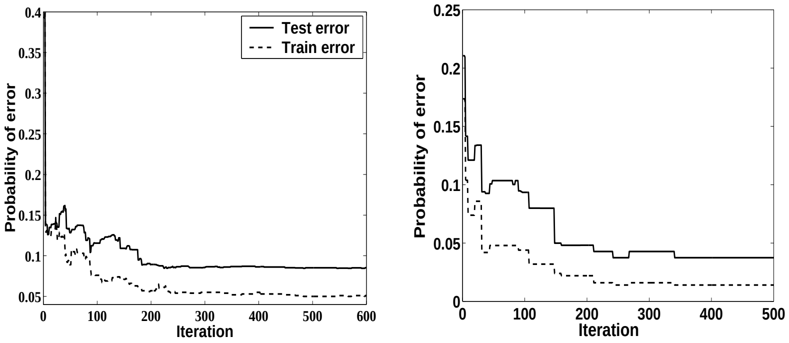
TABLE I

CLASSIFICATION RESULTS (IN %) FOR NAIVE BAYES,TAN, EM-CBL1 AND STOCHASTIC STRUCTURE SEARCH. XX-L INDICATES LEARNING ONLY WITH THE AVAILABLE LABELED DATA.

Dataset	Train		Test	NB-L	EM-NB	TAN-L	EM-TAN	EM-CBL1	SSS
	# lab	# unlab							
TAN artificial	300	30000	50000	83.4±0.2	59.2±0.2	90.9±0.1	91.9±0.1	N/A	91.1±0.1
Satimage	600	3835	2000	81.7±0.9	77.5±0.9	**83.5±0.8**	81.0±0.9	**83.5±0.8**	83.4±0.8
Shuttle	100	43400	14500	82.4±0.3	76.1±0.4	81.2±0.3	90.5±0.2	91.8±0.2	**96.3±0.2**
Adult	6000	24163	15060	83.9±0.3	73.1±0.4	84.7±0.3	80.0±0.3	82.7±0.3	**85.0±0.3**
Chess	150	1980	1060	79.8±1.2	62.1±1.5	**87.0±1.0**	71.2±1.4	81.0±1.2	76.0±1.3

datasets to learn the structure, the resultant networks still suffer from performance degradation when learned with unlabeled data.

Illustrating the iterations of the SSS algorithm, Figure 4(b) shows the changes in error for the shuttle datasets. The Bayesian network structure learned with the SSS algorithm for the Shuttle database is shown in Figure 5



(a) (b)

Fig. 4. Train and test error during the structure search for the artificial data (a) and shuttle data (b) for the labeled and unlabeled data experiments.

IV. LEARNING BAYESIAN NETWORK CLASSIFIERS FOR REAL APPLICATIONS

The experiments in the previous section discussed commonly used machine learning datasets. In this section and the following, we discuss two real applications that could benefit from the use of unlabeled data.

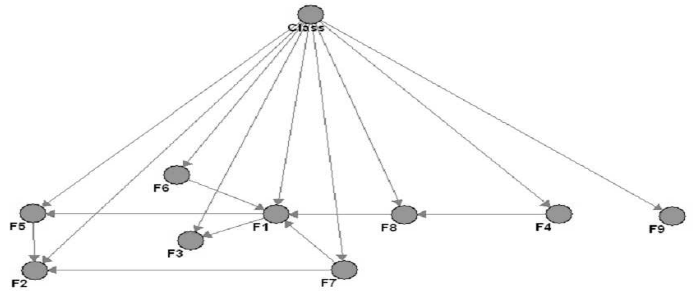


Fig. 5. Bayesian network structure learned for the Shuttle database

We start with facial expression recognition.

A. Facial Expression Recognition using Bayesian Network Classifiers

Since the early 1970s, Paul Ekman and his colleagues have performed extensive studies of human facial expressions [42] and found evidence to support universality in facial expressions. These "universal facial expressions" are those representing happiness, sadness, anger, fear, surprise, and disgust. Ekman's work inspired many researchers to analyze facial expressions by means of image and video processing. By tracking facial features and measuring the amount of facial movement, they attempt to categorize different facial expressions. Recent work on facial expression analysis and recognitionhas used these "basic expressions" or a subset of them. In [43], Pantic and Rothkrantz provide an in depth review of many of the research done in automatic facial expression recognition in recent years.

One of the challenges facing researchers attempting to design facial expression recognition systems is the relatively small amount of available labeled data. Construction and labeling of a good database of images or videos of facial expressions requires expertise, time, and training of subjects. Only a few such databases are available, such as the Cohn-Kanade database [44]. However, collecting, without labeling, data of humans displaying expressions is not as difficult. Therefore, it is beneficial to use classifiers that can be learned with a combination of some labeled data and a large amount of unlabeled data. As such we use (generative) Bayesian network classifiers.

We have developed a real time facial expression recognition system [45]. The system uses a model based non-rigid face tracking algorithm [46] to extract motion features (seen in Figure 8(a)) that serve as input to a Bayesian network classifier used for recognizing the different facial expressions. There are two main motivations for using Bayesian network classifiers in this problem. The first is the ability to learn with unlabeled data and infer the class label even when some of the features are missing (e.g., due to failure in tracking because of occlusion). The second motivation is that it is possible to extend the system to fuse other modalities, such as audio, in a principled way by simply adding subnetworks representing the audio features. A snap shot of the system, with the face tracking and recognition result is shown in Figure 6.

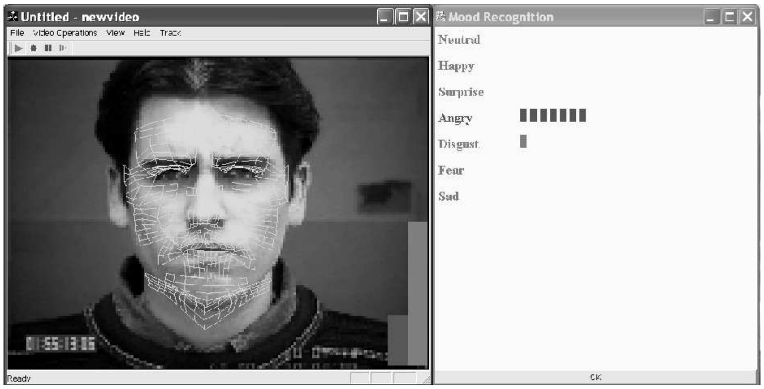


Fig. 6. A snap shot of our realtime facial expression recognition system. On the right side is a wireframe model overlayed on a face being tracked. On the left side the correct expression, Angry, is detected (the bars show the relative probability of Angry compared to the other expressions).

A.1 Experimental Design

We use two different databases, a database collected by Chen and Huang [47] and the Cohn-Kanade AU code facial expression database [44]. The first is a database of subjects that were instructed to display facial expressions corresponding to the six types of emotions. All the tests of the algorithms are performed on a set of five people, each one displaying six sequences of each one of the six emotions, starting and ending at the Neutral expression. The video sampling rate was 30 Hz, and a typical emotion sequence is about 70 samples long ($\sim$2s). Figure 7(upper row) shows one frame of each subject.

The Cohn-Kanade database [44] consists of expression sequences of subjects, starting from a Neutral expression and ending in the peak of the facial expression. There are 104 subjects in the database. Because for some of the subjects, not all of the six facial expressions sequences were available to us, we used a subset of 53 subjects, for which at least four of the sequences were available. For each subject there is at most one sequence per expression with an average of 8 frames for each expression. Figure 7(lower row) shows some examples used in the experiments. A summary of both databases is presented in Table II. We measure the accuracy with respect to the classification result of each frame, where each frame in the video sequence was manually labeled to one of the expressions (including Neutral). This manual labeling can introduce some ’noise’ in our classification because the boundary between Neutral and the expression of a sequence is not necessarily optimal, and frames near this boundary might cause confusion between the expression and the Neutral.

TABLE II

SUMMARY OF THE DATABASES

Database	# of Subjects	Overall # of sequences per expression	# of sequences per subject per expression	average # of frames per expression
Chen-Huang DB	5	30	6	70
Cohn-Kanade DB	53	53	1	8

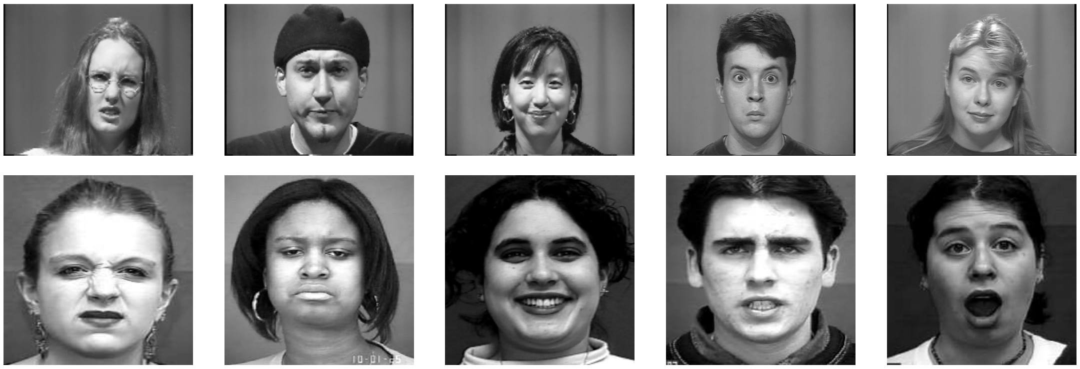


Fig. 7. Examples of images from the video sequences used in the experiment. Top row shows subjects from the Chen-Huang DB, bottom row shows subjects from the Cohn-Kanade DB (printed with permission from the researchers).

A.2 Experimental results with labeled data

We start with experiments using all our labeled data. This can be viewed as an upper bound on the performance of the classifiers trained with most of the labels removed. For the labeled only case, we also compare results with training of an artificial Neural network (ANN) so as to test how Bayesian network classifiers compare with different kind of classifiers for this problem. We perform person independent tests by partitioning the data such that the sequences of some subjects are used as the test sequences and the sequences of the remaining subjects are used as training sequences. Table III shows the recognition rate of the test for all classifiers. The classifier learned with the SSS algorithm outperforms both the NB and TAN classifiers, while ANN do not perform well compared to all the others.

TABLE III

RECOGNITION RATE (%) FOR PERSON-INDEPENDENT TEST.

	NB	TAN	SSS	ANN
Chen-Huang Database	71.78	80.31	83.62	66.44
Cohn-Kandade Database	77.70	80.40	81.80	73.81

It is also informative to look at the structures that were learned from data. Figure 8 shows two learned tree structure of the features (our Motion Units) one learned using the Cohn-Kanade database and the second from the Chen-Huang database. The arrows are from parents to children MUs. In both tree structures we see that the algorithm produced structures in which the bottom half of the face is almost disjoint from the top portion, except for a link between MU9 and MU8 in the first and a weak link between MU4 and MU11 in the second.

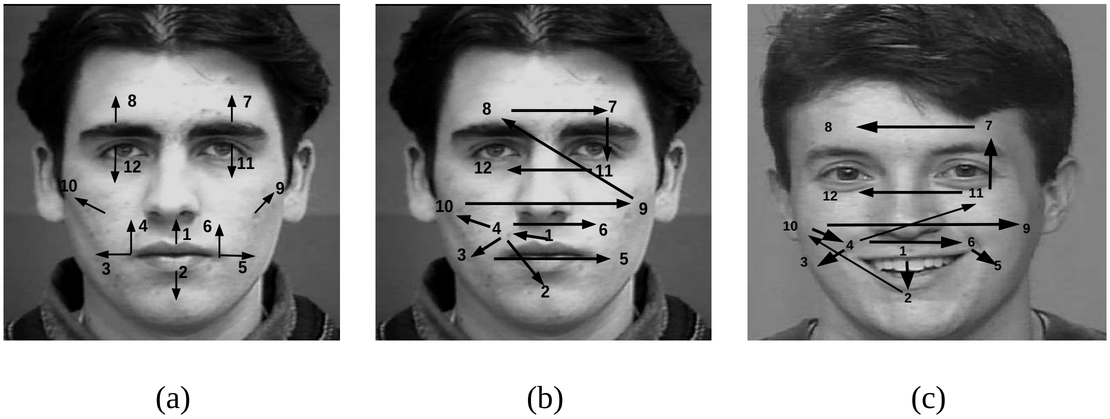


Fig. 8. (a) Motion units extracted from face tracking, (b+c) Two learned TAN structures for the facial features, (b) using the Cohn-Kanade database, (c) using the Chen-Huang database.

A.3 Experiments with labeled and unlabeled data

We perform person-independent experiments with labeled and unlabeled data. We first partition the data to a training set and a test set (2/3 training, 1/3 for testing),and choose by random a portion of the training set and remove the labels. This procedure ensures that the distribution of the labeled and the unlabeled sets are the same.

TABLE IV

CLASSIFICATION RESULTS FOR FACIAL EXPRESSION RECOGNITION WITH LABELED AND UNLABELED DATA.

Dataset	Train		Test	NB-L	EM-NB	TAN-L	EM-TAN	EM-CBL1	SSS
	# lab	# unlab							
Cohn-Kanade	200	2980	1000	72.5±1.4	69.1±1.4	72.9±1.4	69.3±1.4	66.2±1.5	**74.8**±**1.4**
Chen-Huang	300	11982	3555	71.3±0.8	58.5±0.8	72.5±0.7	62.9±0.8	65.9±0.8	**75.0**±**0.7**

We then train Naive Bayes and TAN classifiers, using just the labeled part of the training data and the combination of labeled and unlabeled data. We also use the SSS and the EM-CBL1 algorithms to train a classifier using both labeled and unlabeled data (we do not search for the structure with just the labeled part because it is too small for performing a full structure search).

Table IV shows the results of the experiments. We see that with NB and TAN, when using 200 and 300 labeled samples, adding the unlabeled data degrades the performance of the classifiers, and we would have been better off not using the unlabeled data. We also see that EM-CBL1 performs poorly in both cases. Using the SSS algorithm, we are able to improve the results and utilize the unlabeled data to achieve performance which is higher than using just the labeled data with NB and TAN. The fact that the performance is lower than in the case when all the training set was labeled (about 75% compared to over 80%) implies that the relative value of labeled data is higher than of unlabeled data, as was shown by Castelli [17]. However, had there been more unlabeled data, the performance would be expected to improve.

B. Applying Bayesian Network Classifiers to Face Detection

We apply Bayesian network classifiers to the problem of face detection, with the purpose of showing that using our proposed methods, semi-supervised learning can be used to learn good face detectors. We take an appearance based approach, using the intensity of image pixels as the features for the classifier. For learning and defining the Bayesian network classifiers, we must look at fixed size windows and learn how a face appears in such windows, where we assume that the face appears in most of the window's pixels. The goal of the classifier would be to determine if the pixels in a fixed size window are those of a face or non-face.

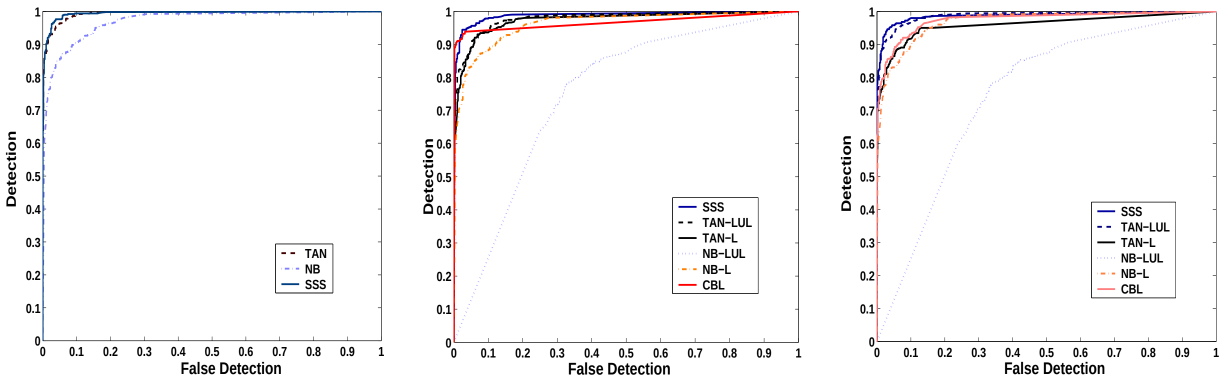
We note that there have been numerous appearance based approaches for face detection, many with considerable success (see Yang et al. [48] for a detailed review on the state-of the-art in face detection). However, there has not been any attempt, to our knowledge, to use semi-supervised learning in face detection. While labeled databases of face images are available, a universally robust face detector is still difficult to construct. The main challenge is that faces appear very different under different lighting conditions, expressions, with or without glasses, facial hair, makeup, etc. A classifier trained with some labeled images and a large number of unlabeled images would enable incorporating many more facial variations without the need to label huge datasets.

In our experiments we used a training set consisting of 2429 faces and 10000 non faces obtained from the MIT CBCL Face database #1 [49]. Each face image is cropped and resampled to a 8x8 window, thus we have a classifier with 64 features. We also randomly rotate and translate the face images to create a training set of 10000 face images. In addition we have available 10000 non-face images. We leave out 1000 images (faces and non-faces) for testing and train the Bayesian network classifier on the remaining 19000. In all the experiments we learn a Naive Bayes, a TAN, and two general generative Bayesian network classifiers, the latter using the EM-CBL1 and the SSS algorithms.

To compare the results of the classifiers, we use the receiving operating characteristic (ROC) curves. The ROC curves show, under different classification thresholds, ranging from 0 to 1, the probability of detecting a face in a face image, $P_D = P(\hat{C} = \mathit{face} | C = \mathit{face})$, against the probability of falsely detecting a face in a non-face image,$P_{FD} = P(\hat{C} = \mathit{face} | C \neq \mathit{face})$.

We first learn using all the training data being labeled. Figure 9(a) shows the resultant ROC curve for this case. The classifier learned with the SSS algorithm outperforms both TAN and NB classifiers, and all perform quite well, achieving about 96% detection rates with a low rate of false alarm.

Next we remove the labels of 95% of the training data (leaving only 475 labeled images) and train the classifiers. Figure 9(b) shows the resultant ROC curve for this case. We see that NB classifier using both labeled and unlabeled data performs very poorly. The TAN based on the 475 labeled images and the TAN based on the labeled and unlabeled images are close in performance, thus there was no significant degradation of performance when adding the unlabeled data. The classifier using all data and the SSS outperforms the rest with an ROC curve close to the best ROC curve in Figure 9(a). Figure 9(c) shows the ROC curve



(a) (b) (c)

Fig. 9. ROC curves showing detection rates of faces compared to false detection of faces of the different (SSS, TAN and NB) classifiers and different ratios of labeled and unlabeled data, (a) with all the data labeled (no unlabeled data), (b) with 95% of the data unlabeled, (c) with 97.5% of the data unlabeled.

with only 250 labeled data used. Again, NB with both labeled and unlabeled performs poorly, while SSS outperforms the other classifiers with no great reduction of performance compared to the two other ROC curves. The experiment shows that using structure search, the unlabeled data was utilized successfully to achieve a classifier almost as good as if all the data was labeled.

V. Summary and Discussion

Using unlabeled data to enhance the performance of classifiers trained with few labeled data has many applications in pattern recognition such as computer vision, HCII, data mining, text recognition and more. To fully utilize the potential of unlabeled data, the abilities and limitations of existing methods must be understood.

The main contributions of this paper can be summarized as follows:

1. We have derived and studied the asymptotic behavior of semi-supervised learning based on maximum likelihood estimation. We presented a detailed analysis of performance degradation from unlabeled data, showing that it is directly related to modeling assumptions, regardless of numerical instabilities or finite sample effects.

2. We discussed the implications of the analysis of semi-supervised learning on Bayesian network classifiers; namely the importance of structure when unlabeled data are used in training. We listed the possible shortcomings of likelihood-based structural learning algorithms when learning classifiers, especially when unlabeled

data are present.

3. We introduced a classification driven structure search algorithm based on Metropolis-Hastings sampling, and showed that it performs well both on fully labeled datasets and on labeled and unlabeled training sets. As a note for practitioners, the SSS algorithm appears to work well for relatively large datasets and difficult classification problems that are represented by complex structures. Large datasets are those where there are enough labeled data for reliable estimation of the empirical error, allowing search for complex structures, and there are enough unlabeled data to reduce the estimation variance of complex structures.

4. We presented our real-time facial expression recognition system using a model-based face tracking algorithm and Bayesian network classifiers. We showed experiments using both labeled and unlabeled data.

5. We presented the use of Bayesian network classifiers for learning to detect faces in images. We note that while finding a good classifier is a major part of any face detection system, there are many more components that need to be designed for such a system to work on natural images (e.g., ability to detect at multi-scales, highly varying illumination, large rotations of faces and partial occlusions). Our goal was to present the first step in designing such a system and show the feasibility of the approach when training with labeled and unlabeled data.

Our discussion of semi-supervised learning for Bayesian networks suggests the following path: when faced with the option of learning Bayesian networks with labeled and unlabeled data, start with Naive Bayes and TAN classifiers, learn with only labeled data and test whether the model is correct by learning with the unlabeled data. If the result is not satisfactory, then SSS can be used to attempt to further improve performance with enough computational resources. If none of the methods using the unlabeled data improve performance over the supervised TAN (or Naive Bayes), either discard the unlabeled data or try to label more data, using active learning for example.

Following our investigation of semi-supervised learning, there are several important open theoretical questions and research directions:

- Is it possible to find necessary and sufficient conditions for performance degradation to occur? Finding such conditions are of great practical significance. Knowing these conditions can lead to the design of new useful tests that will indicate when unlabeled can be used or when they should be discarded, or if a different model should be chosen.

- An important question is whether other semi-supervised learning methods, such as transductive SVM [50] or co-training [51], will exhibit the phenomenon of performance degradation? While no extensive studies have been performed, a few results from the literature suggest that it is a realistic conjecture. Zhang and Oles [2] demonstrated that transductive SVM can cause degradation of performance when unlabeled data are added. Ghani [52] described experiments where the same phenomenon occurred with co-training. If the causes of performance degradation are similar for different algorithms, it should be possible to present a unified theory for semi-supervised learning.
- Are there performance guarantees for semi-supervised learning with finite amounts of data, labeled and unlabeled? In supervised learning such guarantees are studied extensively. PAC and risk minimization bounds help in determining the minimum amount of (labeled) data necessary to learn a classifier with good generalization performance. However, there are no existing bounds on the classification performance when training with labeled and unlabeled data. Finding such bounds can be derived using principals in estimation theory, based on asymptotic covariance properties of the estimator. Other bounds can be derived using PAC theoretical approaches. Existence of such bounds can immediately lead to new algorithms and approaches, better utilizing unlabeled data.
- Can we use the fact that unlabeled data indicates model incorrectness to actively learn better models? The use of active learning seems promising whenever possible, and it might be possible to extend active learning to learn better models, not just enhancement of the parameter estimation.

Additionally, other applications could benefit from the analysis of this work, such as content based image retrieval, text understanding, classification in bio-informatics and more.

In closing, it is possible to view some of the components of this work independently of each other. The theoretical results of Section II do not depend on the choice of probabilistic classifier and can be used as a guide to other choices of classifiers. Structure learning of Bayesian networks is not a topic motivated solely by the use of unlabeled data. Facial expression recognition and face detection could be solved using classifiers other than Bayesian networks. However, this work should be viewed as a combination of all three components; the theory showing the limitations of unlabeled data is used to motivate the design of an algorithm to search for better performing structures of Bayesian networks and finally, the successful application to the real-world problems we were interested in solving by learning with labeled and unlabeled data.

ACKNOWLEDGMENTS

The authors would like to acknowledge Alex Bronstein and Marsha Duro of HP-Labs for their support and discussions on the analysis of semi-supervised learning. Portions of this work have been done in HP-Labs, Palo-Alto. Ira Cohen has been funded by an HP Fellowship. Financial support has also been granted by HP-Brazil through the Eldorado foundation. The authors would like to thank Jilin Tu for his work on the face tracking, Roy Wang and Ming-Hsuan Yang for their help with the face detection database and Jeffery Cohn and Larry Chen for the use of the facial expression databases. We would like to thank Moises Goldschmidt, Ashutosh Garg, Ismênia Blavatsky, and Michael Lew for discussions on various parts of this work. We coded our own Naive Bayes and TAN classifiers in the Java language, using the libraries of the JavaBayes system (freely available at http://www.cs.cmu.edu/˜javabayes).

REFERENCES

[1] B. Shahshahani and D. Landgrebe, "Effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 5, pp. 1087–1095, 1994.

[2] T. Zhang and F. Oles, "A probability analysis on the value of unlabeled data for classification problems," in *International Conference on Machine Learning (ICML)*, pp. 1191–1198, 2000.

[3] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Machine Learning*, vol. 39, no. 2, pp. 103–134, 2000.

[4] R. Bruce, "Semi-supervised learning using prior probabilities and EM." presented at the International Joint Conference of AI Workshop on Text Learning: Beyond Supervision, Seattle, Washington, 2001.

[5] S. Baluja, "Probabilistic modelling for face orientation discrimination: Learning from labeled and unlabeled data," in *Neural Information and Processing Systems (NIPS)*, pp. 854–860, 1998.

[6] R. Kohavi, "Scaling up the accuracy of naive Bayes classifiers: A decision-tree hybrid," in *Proc. Second Int. Conference on Knowledge Discovery and Data Mining*, pp. 202–207, 1996.

[7] I. Cohen, F. G. Cozman, and A. Bronstein, "On the value of unlabeled data in semi-supervised learning based on maximum-likelihood estimation," Tech. Rep. HPL-2002-140, HP-Labs, 2002.

[8] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, California: Morgan Kaufmann, 1988.

[9] A. Garg, V. Pavlovic, J. Rehg, and T. S. Huang, "Multimodal speaker detection using error feedback dynamic Bayesian networks." To be published in Computer Vision and Pattern Recognition 2000.

[10] N. Oliver, E. Horvitz, and A. Garg, "Hierarchical representations for learning and inferring office activity from multimodal information," in *International Conference on Multimodal Interfaces, (ICMI)*, 2002.

[11] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, no. 2, pp. 131–163, 1997.

[12] R. Greiner and W. Zhou, "Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers," in *Proc. of the Annual National Conference on Artificial Intelligence (AAAI)*, pp. 167–173, 2002.

[13] P. Ekman and W. Friesen, *Facial Action Coding System: Investigator's Guide*. Palo Alto: Consulting Psychologists Press, 1978.

[14] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases, University of California, Irvine, Dept. of Information and Computer Sciences," 1998.

[15] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York: Springer Verlag, 1996.

[16] A. Corduneanu and T. Jaakkola, "Continuations methods for mixing heterogeneous sources," in *Uncertainty in Artificial Intelligence (UAI)*, pp. 111–118, 2002.

[17] V. Castelli, *The relative value of labeled and unlabeled samples in pattern recognition*. PhD thesis, Stanford University, Palo Alto, CA, 1994.

[18] J. Ratsaby and S. S. Venkatesh, "Learning from a mixture of labeled and unlabeled examples with parametric side information," in *Proceedings of the Eigth Annual Conference on Computational Learning Theory*, pp. 412–417, 1995.

[19] F. G. Cozman and I. Cohen, "Unlabeled data can degrade classification performance of generative classifiers," in *Fifteenth International Florida Artificial Intelligence Society Conference*, pp. 327–331, 2002.

[20] I. Cohen, *Semisupervised learning of classifiers with application to human-computer interaction*. PhD thesis, University of Illinois at Urbana-Champaign, 2003.

[21] F. G. Cozman and I. Cohen, "The effect of modeling errors in semi-supervised learning of mixture models: How unlabeled data can degrade performance of generative classifiers." http://www.poli.usp.br/p/fabio.cozman/ Publications/lul.ps.gz, 2003.

[22] H. White, "Maximum likelihood estimation of misspecified models," *Econometrica*, vol. 50, pp. 1–25, January 1982.

[23] S. Ganesalingam and G. J. McLachlan, "The efficiency of a linear discriminant function based on unclassified initial samples," *Biometrika*, vol. 65, pp. 658–662, December 1978.

[24] S. W. Ahmed and P. A. Lachenbruch, "Discriminant analysis when scale contamination is present in the initial sample," in *Classification and Clustering*, (New York), pp. 331–353, Academic Press Inc., 1977.

[25] G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. New York: John Wiley and Sons Inc., 1992.

[26] J. H. Friedman, "On bias, variance, 0/1-loss, and the curse-of-dimensionality," *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 55–77, 1997.

[27] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[28] A. Garg and D. Roth, "Understanding probabilistic classifiers," in *European Conference on Machine Learning*, pp. 179–191, 2001.

[29] C. K. Chow and C. N. Liu, "Approximating discrete probability distribution with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[30] M. Meila, *Learning with mixture of trees*. PhD thesis, Massachusetts Institute of Technology, Boston, MA, 1999.

[31] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*. Cambridge: MIT Press, 2nd ed., 2000.

[32] J. Pearl, *Causality: Models, Reasoning, and Inference*. Cambridge: Cambridge University Press, 2000.

[33] J. Cheng, R. Greiner, J. Kelly, D. A. Bell, and W. Liu, "Learning Bayesian networks from data: an information-theory based approach," *Artificial Intelligence Journal*, vol. 137, pp. 43–90, May 2002.

[34] J. Cheng and R. Greiner, "Comparing Bayesian network classifiers," in *Uncertainty in Artificial Intelligence (UAI)*, pp. 101–108, 1999.

[35] T. V. Allen and R. Greiner, "A model selection criteria for learning belief nets: An empirical comparison," in *International Conference on Machine Learning (ICML)*, pp. 1047–1054, 2000.

[36] N. Friedman, "The Bayesian structural EM algorithm," in *Uncertainty in Artificial Intelligence (UAI)*, pp. 129–138, 1998.

[37] N. Friedman and D. Koller, "Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks," in *Uncertainty in Artificial Intelligence (UAI)*, 2000.

[38] D. Madigan and J. York, "Bayesian graphical models for discrete data," *Int. Statistical Review*, vol. 63, no. 2, pp. 215–232, 1995.

[39] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculation by fast computing machines," *Journal of Chemical Physics*, vol. 21, pp. 1087–1092, 1953.

[40] B. Hajek, "Cooling schedules for optimal annealing," *Mathematics of Operational Research*, vol. 13, pp. 311–329, May 1988.

[41] D. Roth, "Learning in natural language," in *International Joint Conference on Artificial Intelligence*, pp. 898–904, 1999.

[42] P. Ekman, "Strong evidence for universals in facial expressions: A reply to Russell's mistaken critique," *Psychological Bulletin*, vol. 115, no. 2, pp. 268–287, 1994.

[43] M. Pantic and L. J. M. Rothkrantz, "Automatic analysis of facial expressions: The state of the art," *PAMI*, vol. 22, no. 12, pp. 1424–1445, 2000.

[44] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Automatic Face and Gesture Recognition (FG'00)*, pp. 46–53, 2000.

[45] I. Cohen, N. Sebe, A. Garg, and T. S. Huang, "Facial expression recognition from video sequences," in *Proceedings of the International Conference on Multimedia and Expo (ICME)*, pp. 121–124, 2002.

[46] H. Tao and T. S. Huang, "Connected vibrations: A modal analysis approach to non-rigid motion tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 735–740, 1998.

[47] L. S. Chen, *Joint processing of audio-visual information for the recognition of emotional expressions in human-computer interaction*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 2000.

[48] M. H. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *PAMI*, vol. 24, no. 1, pp. 34–58, 2002.

[49] *MIT CBCL Face Database #1*. MIT Center For Biological and Computation Learning: http://www.ai.mit.edu/projects/cbcl, 2002.

[50] K. Bennett and A. Demiriz, "Semi-supervised support vector machines," in *Neural Information and Processing Systems (NIPS)*, pp. 368–374, 1998.

[51] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pp. 92–100, 1998.

[52] R. Ghani, "Combining labeled and unlabeled data for multiclass text categorization," in *International Conference on Machine Learning (ICML)*, pp. 187–194, 2002.