

# Learning Bayesian network classifiers for facial expression recognition using both labeled and unlabeled data

Ira Cohen<sup>1</sup>, Nicu Sebe<sup>2</sup>, Fabio G. Cozman<sup>3</sup>, Marcelo C. Cirelo<sup>3</sup>, Thomas S. Huang<sup>1</sup>

<sup>1</sup>Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, IL, USA  
{iracohen, huang}@ifp.uiuc.edu

<sup>2</sup>Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands  
nicu@liacs.nl

<sup>3</sup>Escola Politécnica, Universidade de São Paulo, São Paulo, Brazil  
{fgcozman, marcelo.cirelo}@usp.br

## Abstract

Understanding human emotions is one of the necessary skills for the computer to interact intelligently with human users. The most expressive way humans display emotions is through facial expressions. In this paper, we report on several advances we have made in building a system for classification of facial expressions from continuous video input. We use Bayesian network classifiers for classifying expressions from video. One of the motivating factor in using the Bayesian network classifiers is their ability to handle missing data, both during inference and training. In particular, we are interested in the problem of learning with both labeled and unlabeled data. We show that when using unlabeled data to learn classifiers, using correct modeling assumptions is critical for achieving improved classification performance. Motivated by this, we introduce a classification driven stochastic structure search algorithm for learning the structure of Bayesian network classifiers. We show that with moderate size labeled training sets and large amount of unlabeled data, our method can utilize unlabeled data to improve classification performance. We also provide results using the Naive Bayes (NB) and the Tree-Augmented Naive Bayes (TAN) classifiers, showing that the two can achieve good performance with labeled training sets, but perform poorly when unlabeled data are added to the training set.

## 1. Introduction

Since the early 1970s, Ekman has performed extensive studies of human facial expressions [10, 11] 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 using 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 has used these “basic expressions” or a subset of them (see Pantic and Rothkrantz’s [19] detailed review of many of the research done in recent years). All these methods are similar in that they first extract some features from the images or video, then these features are used as inputs into a classification system, and the outcome is one of the preselected emotion categories. They differ mainly in the features extracted and in the classifiers used to distinguish between the different emotions.

We have developed a real time facial expression recognition system. The system uses a model based non-rigid face tracking algorithm to extract motion features that serve as input to a Bayesian network classifier used for recognizing facial expressions [5]. In our system, as with all other past research in facial expression recognition, learning the classifiers was done using labeled data and supervised learning algorithms. 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 [14]. However, collecting, without labeling, data of humans displaying expressions is not as difficult. Such data is called unlabeled data. It is beneficial to use classifiers that are learnt with a combination of some labeled data and a large amount of unlabeled data. This paper is focused at describing how to learn to classify facial expressions with labeled and unlabeled data, also known as semi-supervised learning.

Bayesian networks, the classifiers used in our system, can be learned with labeled and unlabeled data using maximum likelihood estimation. One of the main questions is whether adding the unlabeled data to the training set improves the classifier’s recognition performance on unseen data. In Section 3 we briefly discuss our recent results

demonstrating that, counter to statistical intuition, when the assumed model of the classifier does not match the true data generating distribution, classification performance could *degrade* as more and more unlabeled data are added to the training set. Motivated by this, we propose in Section 4 a classification driven stochastic structure search (SSS) algorithm for learning the structure of Bayesian network classifiers. We demonstrate the algorithm’s performance using commonly used databases from the UCI repository [2]. In Section 5 we perform experiments with our facial expression recognition system using two databases and show the ability to use unlabeled data to enhance the classification performance, even with a small labeled training set. We have concluding remarks in Section 6.

## 2. Facial Expression Recognition System

We start with a brief description of our real time facial expression recognition system. The system is composed of a face tracking algorithm which outputs a vector of motion features of certain regions of the face. The features are used as inputs to a Bayesian network classifier.

The face tracking we use in our system is based on a system developed by Tao and Huang [22] called the Piecewise Bézier Volume Deformation (PBVD) tracker. This face tracker uses a model-based approach where an explicit 3D wireframe model of the face is constructed. Once the model is constructed and fitted, head motion and local deformations of the facial features such as the eyebrows, eyelids, and mouth can be tracked. The recovered motions are represented in terms of magnitudes of some predefined motion of various facial features. Each feature motion corresponds to a simple deformation on the face, defined in terms of the Bézier volume control parameters. We refer to these motions vectors as Motion-Units (MU’s). The MU’s used in the face tracker are shown in Figure 1(a). The MU’s are used as the basic features for the classification scheme described in the next sections.

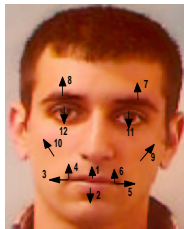


Figure 1: The facial motion measurements

### 2.1. Bayesian network classifiers

We start with a few conventions that are adopted throughout. The goal here is to label an incoming vector of *features* (MUs)  $\mathbf{X}$ . Each instantiation of  $\mathbf{X}$  is a *record*. We assume

that there is a *class variable*  $C$ ; the values of  $C$  are the *labels*, one of the facial expressions. The classifier receives a record  $\mathbf{x}$  and generates a label  $\hat{c}(\mathbf{x})$ . An optimal classification rule can be obtained from the exact distribution  $p(C, \mathbf{X})$ . However, if the distribution is not known, we have to learn it from expert knowledge or data.

For recognizing facial expression using the features extracted from the face tracking system, we consider probabilistic classifiers that represent the a-posteriori probability of the class given the features,  $p(C, \mathbf{X})$ , using Bayesian networks [20]. 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  $\Pi_i$  denotes the parents of  $X_i$  in the graph. 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; otherwise, the structure is *incorrect*. We use maximum likelihood estimation to learn the parameters of the network. When there are missing data in our training set, we use the EM algorithm [9] to maximize the likelihood.

A Bayesian network having the correct structure and parameters is also optimal for classification because the a-posteriori distribution of the class variable is accurately represented. A Bayesian network classifier is a *generative* classifier 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 classifiers and exclude structures that are diagnostic, which cannot be trained using maximum likelihood approaches with unlabeled data [23, 21].

Two examples of generative Bayesian network classifiers are the Naive Bayes (NB) classifier, in which the features are assumed independent given the class, and the Tree-Augmented Naive Bayes classifier (TAN). The NB classifier makes the assumption that all features are conditionally independent given the class label. Although this assumption is typically violated in practice, NB have been used successfully in many classification applications. One of the reasons for the NB success is attributed to the small number of parameters needed to be learnt.

In the structure of the TAN classifier, the class variable is the parent of all the features and each feature has at most one other feature as a parent, such that the resultant graph of the features forms a tree. Using the algorithm presented by Friedman et al. [12], the most likely TAN classifier can be estimated efficiently. When unlabeled data are available, estimating the parameters of the Naive Bayes classifier can be done using the EM algorithm. As for learning the TAN

classifier, we learn the structure and parameters using the EM-TAN algorithm, derived from [16].

We have previously used both the NB and TAN classifiers to perform facial expression recognition [6, 5] with good success. However, we used only labeled data for classification. With unlabeled data we show in our experiments that the limited expressive power of Naive Bayes and TAN causes the use of unlabeled data to degrade the performance of our recognition system. This statement will become clear as we describe the properties of learning with labeled and unlabeled data in the next section.

### 3. Learning a classifier from labeled and unlabeled training data

In this section we discuss properties of classifiers learned with labeled and unlabeled data. In particular, we discuss the possibility that unlabeled data *degrade* classification performance.

Early work proved that unlabeled data lead to improved classification performance, *provided that* the modeling assumptions of the classifier are correct [3, 23]. These have advanced an optimistic view of the labeled-unlabeled problem, where unlabeled data can be profitably used whenever available. However, unlabeled data can also lead to significant degradation in classification performance. A few results in the literature illustrate this possibility. Nigam et al [18] use Naive Bayes classifiers and a large number of features, and report that, when modeling assumptions “are not satisfied, EM may actually degrade rather than improve classifier accuracy” and suggest giving a smaller weight to the unlabeled data. Baluja [1] use unlabeled data to help learn how to determine face orientation. He observed that with Naive Bayes classifiers, unlabeled data sometimes degraded the performance, and proceeded to model the dependencies among the features, finding that such models use better the unlabeled data.

We have conducted an investigation on the effect of unlabeled data and showed that unlabeled data can have deleterious effect when the modeling assumptions are incorrect [8]; here we summarize the main points. We have observed that degradation is not just caused by numerical problems, such as local convergence of the EM algorithm; nor is it just caused by differences between the distribution of labeled data and the distribution of unlabeled data; nor is it just caused by outliers. These explanations do not suffice to clarify why is it that labeled records are routinely seen to improve classification, even in the presence of outliers or incorrect clusters of features, while the same modeling problems lead unlabeled data to degrade classification. This degradation occurs because the asymptotic classification performance of a classifier with incorrect structure can be different when this classifier is learned with fully labeled data and when the classifier is learned with labeled and un-

labeled data. Moreover, we proved that there is a fundamental lack of robustness of maximum likelihood estimators when trained with labeled and unlabeled data under incorrect modeling assumptions.

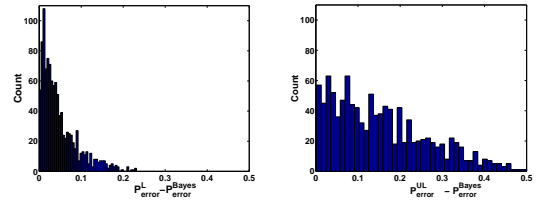


Figure 2: Histogram of classification error bias from the Bayes error rate under incorrect independence assumptions for training with labeled data (left) and training with unlabeled data (right).

Consider Figure 2 which illustrates the differences in classification bias of classifiers learned from labeled and unlabeled data, where the bias is measured from the Bayes error rate. We simulate the asymptotic case of infinite data.<sup>1</sup>

We generated 100 different binary classifiers, each with 4 Gaussian distributed features given the class, and not independent of each other. The parameters of each classifier are: the class prior,  $\eta = p(C = 0)$ , the mean vectors,  $\mu_0, \mu_1 \in \mathbb{R}^4$ , and a common covariance matrix  $S \in \mathbb{R}^{4 \times 4}$ . The Bayes error rate of the classifiers ranged from 0.7–35%, with most being around 10% (the Bayes error was computed analytically using the true parameters of the classifiers).

For each classifier we looked at different combinations of making incorrect independence assumptions, by assuming that features are independent of each other (from one to all features being independent of each other; overall 11 combinations). For example, if we assume that  $x_1$  and  $x_3$  are independent of the rest of the features, the covariance matrix we estimate under this assumption must have the form:

$$\hat{S} = \begin{pmatrix} s_{11} & 0 & 0 & 0 \\ 0 & s_{22} & 0 & s_{24} \\ 0 & 0 & s_{33} & 0 \\ 0 & s_{42} & 0 & s_{44} \end{pmatrix} \quad (1)$$

thus some elements of the covariance matrix are incorrectly forced to be zero.

For each combination we computed the classification error of two classifiers (trained under the independence assumptions): one simulating training with infinite labeled data and a second trained with infinite unlabeled data. For the labeled data case, since the ML estimation is unbiased, the learned parameters are the true priors, the means, and the elements of the covariance matrix that were not forced to be zero. For unlabeled data, we approximated infinity with

<sup>1</sup>Care should be taken when using only unlabeled data in training. As noted by Castelli [3], with unlabeled data it is possible to recover all the parameters of the classifier (under some restrictions, such as identifiability), but a decision on the actual labeling is not possible since we do not know what are the class labels. In the following we assume that we are given this knowledge and therefore are able to perform classification.

100,000 training records (which is very large compared to 25, the largest number of parameters estimated in the experiments). We used EM to learn with unlabeled data, with the starting point being the parameter set of the labeled only classifier, therefore assuring that the difference in the results of the two estimated classifiers do not depend on the starting point of EM.

Over all, we computed 1100 classification errors for the completely labeled case and 1100 for the unlabeled case. From the errors we generated the classification error bias histograms in Figure 2. The histograms show that the classification bias of the labeled based classifiers tends to be more highly concentrated closer to 0 compared to the unlabeled based classifiers. We also observed that using unlabeled data always resulted in a higher error rate compared to using labeled data. The only exception was when we did not make any incorrect independence assumptions, in which the classifiers trained with unlabeled data achieved the Bayes error rate, as expected. What we understand from these histograms is that when training with labeled data, many classifiers will perform well (although never achieve the optimal Bayes rate). However, classifiers trained with unlabeled data need to be more accurate in their modeling assumptions to achieve good performance and they are a great deal more sensitive to such inaccuracies.

## 4. Learning the structure of Bayesian network classifiers

The conclusion of the previous section indicates the importance of obtaining the correct structure when using unlabeled data in learning the classifier. If the 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 was not proposed by researchers that previously witnessed the performance degradation. Apparently, performance degradation was attributed to unpredictable, stochastic disturbances in modeling assumptions, and not to mistakes in the underlying structure – something that can be detected and fixed.

One attempt to overcome the performance degradation from unlabeled data could be to switch models as soon as degradation is detected. Suppose 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. An interesting idea is to start with a Naive Bayes classifier and, if performance degrades with unlabeled data, switch to a different type of Bayesian network classifier, namely the TAN classifier. If the correct structure can be represented using a TAN structure, this approach will indeed work. However, even the TAN structure is only a small set of all possible structures. Moreover, as the experiments in

the next sections show, switching from NB to TAN does not guarantee that the performance degradation will not occur.

A different approach to overcome performance degradation is to use some standard structure learning algorithm, as there are many such algorithms in the Bayesian network literature [12, 7]. A common goal of many existing methods is to find a structure that best fits the joint distribution of all the variables given the data. Because learning is done with finite datasets, most methods penalize very complex structures that might overfit the data, using for example the minimum description length (MDL) score. The difficulty of structure search is the size of the space of possible structures. With finite amounts of data, algorithms that search through the space of structures maximizing the likelihood, can lead to poor classifiers because the a-posteriori probability of the class variable could have a small effect on the score [12]. Therefore, a network with a higher score is not necessarily a better classifier. Friedman et al [12] further suggest changing the scoring function to focus only on the posterior probability of the class variable, but show that it is not computationally feasible.

The drawbacks of likelihood based structure learning algorithms could be magnified when learning with unlabeled data; the posterior probability of the class has a smaller effect during the search, while the marginal of the features would dominate.

### 4.1. Classification driven stochastic structure search

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*

$$inv_e(S') = \frac{1}{\sum_S \frac{1}{p_{S'}(\hat{c}(X) \neq C)}}, \quad (2)$$

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 [17] 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

Dataset	Training records		# Test	NB-L	NB-LUL	TAN-L	TAN-LUL	SSS-LUL
	# labeled	# unlabeled						
TAN artificial	300	30000	50000	83.41%	59.21%	90.89%	91.94%	91.05%
Shuttle	500	43000	14500	82.44%	76.10%	81.19%	90.22%	96.26%
Satimage	600	3835	2000	81.65%	77.45%	83.54%	81.05%	83.35%
Adult	6000	24862	15060	83.86%	73.11%	84.72%	80.00%	85.04%

Table 1: Classification accuracy for Naive Bayes, TAN, and stochastic structure search: Naive Bayes classifier learned with labeled data only (NB-L), Naive Bayes classifier learned with labeled and unlabeled data (NB-LUL), TAN classifier learned with labeled data only (TAN-L), TAN classifier learned with labeled and unlabeled data (TAN-LUL), stochastic structure search with labeled and unlabeled data (SSS-LUL).

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:

$$\min\left(1, \left(\frac{inv_e(S_{new})}{inv_e(S_t)}\right)^{\frac{1}{T}} \frac{q(S_t|S_{new})}{q(S_{new}|S_t)}\right) = \min\left(1, \left(\frac{p_{error}^t}{p_{error}^{new}}\right)^{\frac{1}{T}} \frac{N_t}{N_{new}}\right) \quad (3)$$

where  $q(S'|S)$  is the transition probability from  $S$  to  $S'$ ,  $T$  is a temperature factor, 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 further creates a Markov chain which is aperiodic and irreducible, thus satisfying the Markov chain Monte Carlo (MCMC) conditions [15]. We summarize our algorithm in Figure 3.

1. Fix the network structure to some initial structure,  $S_0$ .
2. Estimate the parameters of the structure  $S_0$  and compute the probability of error  $p_{error}^0$ .
3. Set  $t = 0$ .
4. 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,  $p_{error}^{new}$ .
  - Accept  $S_{new}$  with probability given in Eq. (3).
  - If  $S_{new}$  is accepted, set  $S_{t+1} = S_{new}$  and  $p_{error}^{t+1} = p_{error}^{new}$  and change  $T$  according to the temperature decrease schedule. Otherwise  $S_{t+1} = S_t$ .
  - $t = t + 1$ .
5. return the structure  $S_j$ , such that  $j = \underset{0 \leq j \leq MaxIter}{\operatorname{argmin}} (p_{error}^j)$ .

Figure 3: Stochastic structure search algorithm (SSS)

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 distribution. The rate of decrease of the temperature determines the rate of convergence. Asymptotically in the number of data,

a logarithmic decrease of  $T$  will guarantee convergence to a global maximum with probability that tends to one [13].

There are two caveats though; first, the logarithmic cooling schedule is very slow, second, we never have access to the true probability of error for each structure - we calculate the classification error from a limited pool of training data (denoted by  $\hat{p}_{error}^S$ ). To avoid the problem of overfitting we can take several approaches. Cross-validation can be performed by splitting the labeled training set to smaller sets. However, this approach can significantly slow down the search, and is suitable only if the labeled training set is moderately large. Instead, we use the multiplicative penalty term derived from structural risk minimization to define a modified error term:

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

where  $h_S$  is the Vapnik-Chervonenkis (VC) dimension of the classifier with structure  $S$ ,  $n$  is the number of training records,  $\eta$  and  $c$  are between 0 and 1. To approximate the VC dimension, we use  $h_S \propto N_S$ , with  $N_S$  the number of (free) parameters in the Markov blanket of the class variable in the network, assuming that all variables are discrete.

To illustrate the performance of SSS algorithm, we performed experiments with some of the UCI datasets and an artificially generated data set (a Bayesian network with TAN structure), using relatively small labeled sets and large unlabeled sets (Table 1). The results using the UCI datasets show, to varying degrees, the ability of SSS to utilize unlabeled data. The most dramatic improvement is seen with the Shuttle dataset. The results with the artificially generated data show that SSS was able to achieve almost the same performance as TAN, which had the advantage of a-priori knowledge of the correct structure. We also see that for both NB and TAN, using unlabeled data can cause performance degradation, therefore the idea of switching between these simple models is not guaranteed to work.

## 5. Facial Expression Recognition Experiments

We test the algorithms for the facial expression recognition system. We initially consider experiments where all the data

is labeled. Then we investigate the effect of using both labeled and unlabeled data.

We use two different databases, one collected by Chen and Huang [4] and the Cohn-Kanade database [14]. The first consists of subjects that were instructed to display facial expressions corresponding to six types of emotions. In the Chen-Huang database there are five subjects. For each subjects there are six video sequences per expression, each sequence starting and ending in the Neutral expression. There are on average 60 frames per expression sequence. The Cohn-Kanade database [14] 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 person there are on average 8 frames for each expression.

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.

### 5.1. Experimental results with labeled data

We start with a person-independent experiment using all the labeled data. For this test we use the sequences of some subjects as test sequences and the sequences of the remaining subjects as training sequences (we leave out one subject in the Chen-Huang database and 10 subjects for the Cohn-Kanade database). This test is repeated five times, each time leaving different subjects out (leave one out cross validation). Table 2 shows the recognition rate of the test for all classifiers. We see that the Naive Bayes classifier performs poorly. However, a significant improvement for both the TAN and the SSS algorithm is obtained, with SSS being significantly better. It should be noted that with a smaller training set, SSS would not have been able to explore many structure and its performance would have probably be the same or worse than NB and TAN.

### 5.2. Experiments with labeled and unlabeled data

We consider now both labeled and unlabeled data in a person-independent experiment. We first partition the data to a training set and a test set and randomly choose 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 2: Recognition rates (%) for person-independent test

	NB	TAN	SSS
Chen-Huang Database	71.78	80.31	<u>83.62</u>
Cohn-Kandade Database	77.70	80.40	<u>81.80</u>

We train Naive Bayes and TAN classifiers, using just the labeled part of the training data and the combination of labeled and unlabeled data. We use the SSS algorithm 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).

We see in Table 3 that with NB and TAN, even when using only 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. Using the SSS algorithm, we are able to improve the results and use 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 (see Table 2) implies that the relative value of labeled data is higher than of unlabeled data, as was shown by Castelli [3]. However, had there been more unlabeled data, the performance would be expected to improve.

## 6. Summary and Discussion

In this work, we presented several advances we made in building a real-time system for classification of facial expressions from continuous video input. The facial expression recognition was done using Bayesian networks classifiers. Collecting labeled data of humans displaying expressions is a difficult task and therefore, we were interested in learning the classifiers with both labeled and unlabeled data. One question we asked was whether adding the unlabeled data to the training set improves the classifier's recognition performance on unseen data. We showed that when incorrect modeling assumptions are used, the unlabeled data could have deleterious effect on the classification performance, while the same unlabeled data, under correct modeling assumptions, are theoretically guaranteed to improve the classification performance. With this result we proposed a classification driven stochastic structure search algorithm for learning the structure of the Bayesian network classifiers. We demonstrated the algorithm's performance using standard databases from the UCI repository. Using moderate size labeled training sets and large amount of unlabeled data, our method was able to utilize unlabeled data to improve classification performance.

We tested our classifiers for facial expression recognition using two databases. We compared the results with two

Table 3: Classification results for facial expression recognition with labeled and unlabeled data.

Dataset	Training records		# Test	NB-L	NB-LUL	TAN-L	TAN-LUL	SSS-LUL
	# labeled	# unlabeled						
Cohn-Kanade	200	2980	1000	72.50%	69.10%	72.90%	69.30%	74.80%
Chen-Huang	300	11982	3555	71.25%	58.54%	72.45%	62.87%	74.99%

other Bayesian network classifiers that have been used in our system: Naive Bayes and TAN networks and we showed that the two can achieve good performance with labeled training sets, but perform poorly when unlabeled data are added to the training set. We showed that by searching for the structure driven by the classification error enables us to use the unlabeled data to improve the classification performance.

In conclusion, our main contributions are as follows. We applied Bayesian network classifiers to the problem of facial expression recognition and we proposed a method that can effectively search for the correct Bayesian network structure focusing on classification. We also stressed the importance of obtaining such a structure when using unlabeled data in learning the classifier. If correct structure is used, the unlabeled data improve the classification, otherwise they can actually degrade the performance. Finally, we integrated the classifiers and the face tracking system to build a real time facial expression recognition system.

## Acknowledgments

We thank Alex Bronstein and Marsha Duro at HP-Labs for proposing the research on labeled-unlabeled data and for many suggestions and comments during the course of the work on this topic, as their help was critical to the results described here. We coded our own classifiers in the Java language, using the libraries of the JavaBayes system (freely available at <http://www.cs.cmu.edu/~javabayes>). This work has been supported in part by the National Science Foundation Grants CDA-96-24396 and IIS-00-85980. The work of Ira Cohen has been supported by a Hewlett Packard fellowship.

## References

- [1] S. Baluja. Probabilistic modelling for face orientation discrimination: Learning from labeled and unlabeled data. In *NIPS*, 1998.
- [2] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
- [3] V. Castelli. *The relative value of labeled and unlabeled samples in pattern recognition*. PhD thesis, Stanford, 1994.
- [4] 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, Dept. of Electrical Engineering, 2000.
- [5] I. Cohen, N. Sebe, L.S. Chen, A. Garg, and T.S. Huang. Facial expression recognition from video sequences: Temporal and static modeling. *to appear in CVIU special issue on face recognition*, 2003.
- [6] I. Cohen, N. Sebe, A. Garg, and T.S. Huang. Facial expression recognition from video sequences. In *ICME*, 2002.
- [7] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:308–347, 1992.
- [8] F.G. Cozman and I. Cohen. Unlabeled data can degrade classification performance of generative classifiers. In *FLAIRS*, 2002.
- [9] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
- [10] P. Ekman. Strong evidence for universals in facial expressions: A reply to Russell’s mistaken critique. *Psychological Bulletin*, 115(2):268–287, 1994.
- [11] P. Ekman and W.V. Friesen. *Facial Action Coding System: Investigator’s Guide*. Consulting Psychologists Press, 1978.
- [12] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2):131–163, 1997.
- [13] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of operational research*, 13:311–329, 1988.
- [14] T. Kanade, J. Cohn, and Y. Tian. Comprehensive database for facial expression analysis, 2000.
- [15] D. Madigan and J. York. Bayesian graphical models for discrete data. *Int. Statistical Review*, 63:215–232, 1995.
- [16] M. Meila. *Learning with mixture of trees*. PhD thesis, Massachusetts Institute of Technology, 1999.
- [17] 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*, 21:1087–1092, 1953.
- [18] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39:103–134, 2000.
- [19] M. Pantic and L.J.M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *PAMI*, 22(12):1424–1445, 2000.
- [20] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.
- [21] M. Seeger. Learning with labeled and unlabeled data. Technical report, Edinburgh University, 2001.
- [22] H. Tao and T.S. Huang. Connected vibrations: A modal analysis approach to non-rigid motion tracking. In *CVPR*, pages 735–740, 1998.
- [23] T. Zhang and F. Oles. A probability analysis on the value of unlabeled data for classification problems. In *ICML*, 2000.