

Nicu Sebe · Ira Cohen · Fabio G. Cozman ·  
Theo Gevers · Thomas S. Huang

# Learning probabilistic classifiers for human–computer interaction applications

Published online: 10 May 2005  
© Springer-Verlag 2005

**Abstract** Human–computer interaction (HCI) lies at the crossroads of many scientific areas including artificial intelligence, computer vision, face recognition, motion tracking, etc. It is argued that to truly achieve effective human–computer intelligent interaction, the computer should be able to interact naturally with the user, similar to the way HCI takes place. In this paper, we discuss training probabilistic classifiers with labeled and unlabeled data for HCI applications. We provide an analysis that shows under what conditions unlabeled data can be used in learning to improve classification performance, and we investigate the implications of this analysis to a specific type of probabilistic classifiers, Bayesian networks. Finally, we show how the resulting algorithms are successfully employed in facial expression recognition, face detection, and skin detection.

**Keywords** Semisupervised learning · Bayesian networks · Face detection · Facial expression recognition · Skin detection

---

## 1 Introduction

Recent years have seen a growing interest in improving all aspects of the interaction between humans and computers with the clear goal of achieving a natural interaction, simi-

lar to the way human–human interaction (HCI) takes place. Humans interact with each other mainly through speech, but also through body gestures, to emphasize a certain part of the speech and display of emotions. As a consequence, the new interface technologies are steadily driving toward accommodating information exchanges via the natural sensory modes of sight, sound, and touch. In face-to-face exchange, humans employ these communication paths simultaneously and in combination, using one to complement and enhance another. The exchanged information is largely encapsulated in this natural, multimodal format. Typically, conversational interaction bears a central burden in human communication, with vision, gaze, expression, and manual gestures often contributing critically, as well as frequently embellishing attributes such as emotion, mood, attitude, and attentiveness. But the roles of multiple modalities and their interplay remain to be quantified and scientifically understood. What is needed is a science of human–computer communication that establishes a framework for multimodal “language” and “dialog,” much like the framework we have evolved for spoken exchange.

Another important aspect is the development of human-centered information systems. The most important issue here is how to achieve synergism between man and machine. The term “human-centered” is used to emphasize the fact that, although all existing information systems were designed with human users in mind, many of them are far from being user friendly. What can the scientific/engineering community do to effect a change for the better?

Information systems are ubiquitous in all human endeavors including scientific, medical, military, transportation, and consumer. Individual users use them for learning, searching for information (including data mining), doing research (including visual computing), and authoring. Multiple users (groups of users and groups of groups of users) use them for communication and collaboration, and either single or multiple users use them for entertainment. An information system consists of two components: computer (data/knowledge base and information processing engine) and humans. It is the intelligent interaction between the two

---

N. Sebe (✉) · T. Gevers  
Faculty of Science, University of Amsterdam, The Netherlands  
E-mail: {nicu, gevers}@science.uva.nl

I. Cohen  
HP Labs, Palo Alto, CA, USA  
E-mail: iracohen@hp.com

F. G. Cozman  
Escola Politécnica, Universidade de São Paulo, São Paulo, Brazil  
E-mail: fgcozman@usp.br

T. S. Huang  
Beckman Institute, University of Illinois at Urbana-Champaign, IL,  
USA  
E-mail: huang@ifp.uiuc.edu

that we address in this paper. To do this, in what follows we present in detail three essential components of such an information system for HCI: facial emotion recognition, face detection, and skin detection.

Skin is arguably the most widely used primitive in human image processing research, with applications ranging from face detection [66] and person tracking [55] to pornography filtering [4, 24]. We are especially interested in skin detection as a cue for detecting people (and their faces) in real-world photographs and live videos. Many of the recent applications designed for human–computer intelligent interaction use the human face as an input. Systems that perform face tracking for various applications, facial expression recognition, and pose estimation of faces all rely on detection of human faces in the video frames [48, 49]. Human beings possess and express emotions in everyday interactions with others. Emotions are often reflected on the face, in hand and body gestures, and in the voice to express our feelings or likings [46]. While a precise, generally agreed upon definition of emotion does not exist, it is undeniable that emotions are an integral part of our existence. Facial expressions and vocal emotions are commonly used in everyday human-to-human communication, as one smiles to show greeting, frowns when confused, or raises one’s voice when enraged. People infer a great deal from perceived facial expressions: “You look tired” or “You seem happy.” The fact that we understand emotions and know how to react to other people’s expressions greatly enriches the interaction and defines us as human beings.

Maybe no movie of modern time has explored the definition of what it means to be human better than *Blade Runner*. The Tyrell Corporation’s motto, “More human than human,” serves as the basis for exploring the human experience through true humans and created humans, or replicants. Replicants are androids that were built to look like humans and to work or fight their wars. In time, they began to acquire emotions (so much like humans), and it became difficult to tell them apart. With emotions they began to feel oppressed, and many of them became dangerous and committed acts of extreme violence to be free. Fortunately, Dr. Elden Tyrell, the creator of the replicants, installed a built-in safety feature in these models: a 4-year life span.

It is evident from the above story that it is not sufficient for a machine (computer) to look like a human (e.g., have skin, face and facial features, limbs, etc). Something else is also essential: the ability to acquire emotions. Moreover, the machine should learn to recognize faces and to understand the emotions to be able to have a humanlike interaction with its human counterpart. This paper tries to make only a small dent in this huge task of providing computers with the ability to understand humans.

---

## 2 Background

We present first the current research and the challenges we all face in achieving an automatic HCI system. As we men-

tioned in the previous section, we focus on three components: emotion recognition, face detection, and skin detection.

### 2.1 Emotion recognition research

In many important HCI applications such as computer-aided tutoring and learning, it is highly desirable (even mandatory) that the response of the computer take into account the emotional or cognitive state of the human user. Emotions are displayed by visual, vocal, and other physiological means. There is a growing amount of evidence showing that emotional skills are part of what is called “intelligence” [28]. Computers today can recognize much of what is said and, to some extent, who said it. But they are almost completely “in the dark” when it comes to how things are said, the affective channel of information. This is true not only in speech but also in visual communications despite the fact that facial expressions, posture, and gestures communicate some of the most critical information: how people feel. Affective communication explicitly considers how emotions can be recognized and expressed during HCI.

The most expressive way humans display emotion is through facial expressions. Humans detect and interpret faces and facial expressions in a scene with little or no effort. Still, developing an automated system that accomplishes this task is rather difficult. There are several related problems: detection of an image segment as a face, extraction of the facial expression information, and classification of the expression (e.g., in emotion categories). A system that performs these operations accurately and in real time would be a major step forward in achieving humanlike interaction between human and machine.

Since the early 1970s Paul Ekman and his colleagues have performed extensive studies of human facial expressions [20]. They found evidence to support universality in facial expressions. These “universal facial expressions” are those representing happiness, sadness, anger, fear, surprise, and disgust. They studied facial expressions in different cultures, including preliterate cultures, and found much commonality in the expression and recognition of emotions on the face. However, they observed differences in expressions as well and proposed that facial expressions are governed by “display rules” in different social contexts.

Ekman and Friesen [21] developed the Facial Action Coding System (FACS) to code facial expressions where movements on the face are described by a set of action units (AUs). Each AU has some related muscular basis. This system of coding facial expressions is done manually by following a set of prescribed rules. The inputs are still images of facial expressions, often at the peak of the expression. This process is very time consuming.

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 recognition has used these “basic expressions” or a subset of them. The two recent surveys in the area [23, 45] provide an in-depth review of much of the research done in automatic facial expression recognition in recent years.

Work in computer-assisted quantification of facial expressions did not start until the 1990s. Black and Yacoob [2] used local parameterized models of image motion to recover nonrigid motion. Once recovered, these parameters were used as inputs to a rule-based classifier to recognize the six basic facial expressions. Rosenblum et al. [51] computed optical flow of regions on the face, then applied a radial basis function network to classify expressions. Essa and Pentland [22] used an optical-flow-region-based method to recognize expressions. Donato et al. [18] tested different features for recognizing facial AUs and inferring the facial expression in a frame. Nefian and Hayes [40] proposed an embedded hidden Markov model (HMM) approach for face recognition that uses an efficient set of observation vectors based on the DCT coefficients. Oliver et al. [42] used lower face tracking to extract mouth shape features and used them as inputs to an HMM-based facial expression recognition system (recognizing neutral, happy, sad, and an open mouth). Chen [9] used a suite of static classifiers to recognize facial expressions, reporting on both person-dependent and person-independent results. Cohen et al. [14] describe classification schemes for facial expression recognition in two types of settings: dynamic and static classification. In the static setting, the authors learned the structure of Bayesian network classifiers using as input 12 motion units given by a face tracking system for each frame in a video. For the dynamic setting, they used a multilevel HMM classifier that combines the temporal information and allows one not only to classify video segments with the corresponding facial expressions, as in the previous works on HMM-based classifiers, but also to automatically segment an arbitrary long sequence to the different expression segments without resorting to heuristic methods of segmentation.

These methods are similar in that they first extract some features from the images, then use these features as inputs into a classification system, and the outcome is one of the preselected emotion categories. They differ mainly in the features extracted from the video images and in the classifiers used to distinguish between the different emotions.

## 2.2 Face detection

Images containing a face are essential to intelligent vision-based HCI. The rapidly expanding research in face processing is based on the premise that information about a user’s identity, state, and intention can be extracted from images and that computers can react accordingly, e.g., by observing a person’s facial expression. Given an arbitrary image, the goal of face detection is to automatically locate a human face in an image or video, if it is present. Face detection in a general setting is a challenging problem for various reasons. The first set of reasons are inherent: there

are many types of faces, with different colors, texture, sizes, etc. In addition, the face is a nonrigid object that can change its appearance. The second set of reasons are environmental: changing lighting, rotations, translations, and scales of faces in natural images.

To solve the problem of face detection, two main approaches can be taken. The first is a model-based approach, where a description of what is a human face is used for detection. The second is an appearance-based approach, where we learn what faces are directly from their appearance in images. In this work, we focus on the latter.

There have been numerous appearance-based approaches. We list a few from recent years and refer to the reviews of Yang et al. [66] and Hjelmas and Low [31] for further details. Rowley et al. [52] and Kouzani [35] used neural networks to detect faces in images by training from a corpus of face and nonface images. Colmenarez and Huang [16] used maximum entropic discrimination between faces and nonfaces to perform maximum likelihood classification, which was used for a real-time face tracking system. Yang et al. [65] used SNoW-based classifiers to learn the face and nonface discrimination boundary on natural face images. Others used support vector machines [30]. Wang et al. [62] learned a minimum spanning weighted tree for learning pairwise dependency graphs of facial pixels, followed by a discriminant projection to reduce complexity. Viola and Jones [61] used boosting and a cascade of classifiers for face detection.

Very relevant to our work is the research of Schneiderman [54], who learns a sparse structure of statistical dependencies for several object classes including faces. While analyzing such dependencies can reveal useful information, we go beyond the scope of Schneiderman’s work and present a framework that not only learns the structure of a face but also allows the use of unlabeled data in classification.

Face detection provides interesting challenges to the underlying pattern classification and learning techniques. When a raw or filtered image is considered as input to a pattern classifier, the dimension of the space is extremely large (i.e., the number of pixels in normalized training images). The classes of face and nonface images are decidedly characterized by multimodal distribution functions, and effective decision boundaries are likely to be nonlinear in the image space. To be effective, the classifiers must be able to extrapolate from a modest number of training samples.

## 2.3 Skin detection

The automated detection and tracking of humans in computer vision necessitates improved modeling of human skin appearance. Skin detection is largely used in applications ranging from face detection [60, 66] and person tracking [55] to pornography filtering [4, 24]. The main challenge is to make skin detection robust to the large variations in appearance that can occur. Skin appearance changes in color

and shape are often affected by occlusion (clothing, hair, eye glasses, etc.). Moreover, changes in intensity, color, and location of light sources affect skin appearance. Other objects in the scene may cast shadows or reflect additional light and so forth. Finally, many other objects are easily confused with skin: certain types of wood, copper, sand as well as clothes often have skinlike colors.

Research has been performed on the detection of human skin pixels in color images and on the discrimination between skin pixels and nonskin pixels by use of various statistical color models [32]. Saxe and Foulds [53] proposed an iterative skin identification method that uses histogram intersection in HSV color space. An initial patch of skin color pixels, called the control seed, is chosen by the user and used to initiate the iterative algorithm. To detect skin color regions, their method moves through the image, one patch at a time, and presents a control histogram and current histogram from the image for comparison using the histogram intersection. If the match score is greater than a threshold, the current patch is classified as being skin color. In contrast to the nonparametric methods mentioned above, Gaussian density functions [64] and a mixture of Gaussians [7, 38] are often used to model skin color. The parameters in a unimodal Gaussian distribution are often estimated using maximum likelihood. The motivation for using a mixture of Gaussians is based on the observation that the color histogram for the skin of people with different ethnic backgrounds does not form a unimodal distribution but rather a multimodal distribution. The parameters in a mixture of Gaussians are usually estimated using an EM algorithm. Recently, Jones and Rehg [33] conducted a large-scale experiment in which nearly 1 billion labeled skin-tone pixels were collected (in normalized RGB color space). Comparing the performance of histogram and mixture models for skin detection, they found histogram models to be superior in accuracy and computational cost.

### 3 Learning classifiers for human–computer interaction

Many pattern recognition and HCI applications require the design of classifiers. Classification is the task of systematic arrangement in groups or categories according to some set of observations, e.g., classifying images as those containing human faces and those that do not or classifying individual pixels as being skin or nonskin. Classification is a natural part of daily human activity and is performed on a routine basis. One of the tasks in machine learning has been to give computers the ability to perform classification in different problems. In machine classification, a classifier is constructed that takes as input a set of observations (such as images in the face detection problem) and outputs a prediction of the class *label* (e.g., face or no face). The mechanism that performs this operation is the *classifier*.

We are interested in probabilistic classifiers, in which the observations and class are treated as random variables

and a classification rule is derived using probabilistic arguments (e.g., if the probability of an image being a face given that we observed two eyes, nose, and mouth in the image is higher than some threshold, classify the image as a face). We consider two aspects. First, most of the research mentioned in the previous section tried to classify each observable independently of the others. We want to take a different approach: can we learn the dependencies (the structure) between the observables (e.g., the pixels in an image patch)? Can we use this structure for classification? To achieve this we use Bayesian networks. Bayesian networks can represent joint distributions in an intuitive and efficient way; as such, Bayesian networks are naturally suited for classification. Second, we are interested in using a framework that allows for the usage of labeled and unlabeled data (also called semisupervised learning). The motivation for semisupervised learning stems from the fact that labeled data are typically much harder to obtain compared to unlabeled data. For example, in facial expression recognition it is easy to collect videos of people displaying emotions, but it is very tedious and difficult to link the video to the corresponding expressions. Bayesian networks are very well suited for this task: they can be learned with labeled and unlabeled data using maximum-likelihood estimation.

Is there value in unlabeled data in supervised learning of classifiers? This fundamental question has been increasingly discussed in recent years, with a general optimistic view that unlabeled data hold great value. Due to an increasing number of applications and algorithms that successfully use unlabeled data [1, 3, 6, 27, 41, 57, 58] and magnified by theoretical issues over the value of unlabeled data in certain cases [8, 43, 50], semisupervised learning is seen optimistically as a learning paradigm that can relieve the practitioner of the need to collect many expensive labeled training data. However, several disparate empirical evidences in the literature suggest that there are situations in which the addition of unlabeled data to a pool of labeled data causes degradation of the classifier’s performance [1, 6, 41, 58], in contrast to improvement of performance when adding more labeled data. Intrigued by these discrepancies, we performed extensive experiments, reported in [12, 15]. Our experiments suggested that performance degradation can occur when the assumed classifier’s model is incorrect. Such situations are quite common as one rarely knows whether the assumed model is an accurate description of the underlying true data-generating distribution. More details are given below.

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*. Let  $P(C, \mathbf{X})$  be the *true* joint distribution of the class and features from which any sample of some (or all) of the variables from the set  $\{C, \mathbf{X}\}$  is drawn, and let  $p(C, \mathbf{X})$  be the density distribution associated with it. We want to build *classifiers* that receive a sample  $\mathbf{x}$  and output either of the values of  $C$ .

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 parametrical model  $p(C, \mathbf{X} | \theta)$  is adopted. An estimate of  $\theta$  is denoted by  $\hat{\theta}$ , and we denote throughout by  $\hat{\theta}^*$  the asymptotic value of  $\hat{\theta}$ . 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.” We use “estimation bias” loosely to mean the expected difference between  $p(C, \mathbf{X})$  and the estimated  $p(C, \mathbf{X} | \hat{\theta})$ .

We base our analysis on the work of White [63] on the properties of maximum-likelihood estimators without assuming model correctness. White [63] showed that under suitable regularity conditions, maximum-likelihood estimators converge to a parameter set  $\theta^*$  that minimizes the Kullback–Leibler (KL) distance between the assumed family of distributions,  $p(Y | \theta)$ , and the true distribution,  $p(Y)$ . White also showed 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) approaches infinity.  $C_Y(\theta)$  is a covariance matrix equal to  $A_Y(\theta)^{-1}B_Y(\theta)A_Y(\theta)^{-1}$ , evaluated at  $\theta^*$ , where  $A_Y(\theta)$  and  $B_Y(\theta)$  are matrices whose  $(i, j)$ th element ( $i, j = 1, \dots, d$ , where  $d$  is the number of parameters) is given by:

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

Using these definitions, we obtain the following theorem.

**Theorem 1** *Consider supervised learning where samples are randomly labeled with probability  $\lambda$ . Adopt the regularity conditions in Theorems 3.1, 3.2, 3.3 from [63], 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  $\theta^*$ , the limiting value of maximum likelihood estimates, is:*

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

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  $N \rightarrow \infty$ , where  $C_{\lambda}(\theta)$  is given by:

$$\begin{aligned} C_{\lambda}(\theta) &= A_{\lambda}(\theta)^{-1}B_{\lambda}(\theta)A_{\lambda}(\theta)^{-1} \text{ with,} \\ A_{\lambda}(\theta) &= (\lambda A_{(C, \mathbf{X})}(\theta) + (1 - \lambda)A_{\mathbf{X}}(\theta)) \text{ and} \\ B_{\lambda}(\theta) &= (\lambda B_{(C, \mathbf{X})}(\theta) + (1 - \lambda)B_{\mathbf{X}}(\theta)), \end{aligned} \quad (2)$$

evaluated at  $\theta^*$ .  $\square$

Expression 1 indicates that semisupervised learning can be viewed asymptotically as a “convex” combination of supervised and unsupervised learning. The objective function for semisupervised 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  $\theta_{\lambda}^*$  the value of  $\theta$  that maximizes Eq. 1 for a given  $\lambda$ . Then,  $\theta_1^*$  is the asymptotic estimate of  $\theta$  for supervised learning, denoted by  $\theta_{rml}^*$ . Likewise,  $\theta_0^*$  is the

asymptotic estimate of  $\theta$  for *unsupervised* learning, denoted by  $\theta_u^*$ .

The asymptotic covariance matrix is positive definite as  $B_Y(\theta)$  is positive definite,  $A_Y(\theta)$  is symmetric for any  $Y$ , and

$$\theta A(\theta)^{-1}B_Y(\theta)A(\theta)^{-1}\theta^T = w(\theta)B_Y(\theta)w(\theta)^T > 0,$$

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 what follows, we show this view is valid when the model is correct and that it is not always valid when the model is incorrect.

### 3.1 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  $\theta_{\lambda}^* = \theta_u^* = \theta_{\top}$  given identifiability. Thus  $\theta_{\lambda}^* = \theta_{\top}$  for any  $0 \leq \lambda \leq 1$ .

Additionally, using White’s results [63],  $A(\theta_{\lambda}^*) = -B(\theta_{\lambda}^*) = \mathbf{I}(\theta_{\lambda}^*)$ , where  $\mathbf{I}(\cdot)$  denotes the Fisher information matrix. Thus, the Fisher information matrix can be written as:

$$\mathbf{I}(\theta) = \lambda \mathbf{I}_l(\theta) + (1 - \lambda) \mathbf{I}_u(\theta), \quad (3)$$

which matches the derivations made by Zhang and Oles [67]. The significance of Eq. 3 is that it allows the use of the Cramer–Rao lower bound (CRLB) on the covariance of a consistent estimator:

$$\text{Cov}(\hat{\theta}_N) \geq \frac{1}{N} (\mathbf{I}(\theta))^{-1}, \quad (4)$$

where  $N$  is the number of data (both labeled and unlabeled) and  $\text{Cov}(\hat{\theta}_N)$  is the estimator’s covariance matrix with  $N$  samples.

Consider the Taylor expansion of the classification error around  $\theta_{\top}$ , as suggested by Shahshahani and Landgrebe [58], linking the decrease in variance associated with unlabeled data to a decrease in classification error, and assume the existence of necessary derivatives:

$$\begin{aligned} \mathbf{e}(\hat{\theta}) &\approx \mathbf{e}_B + \left. \frac{\partial \mathbf{e}(\theta)}{\partial \theta} \right|_{\theta_{\top}} (\hat{\theta} - \theta_{\top}) \\ &\quad + \frac{1}{2} \text{tr} \left( \left. \frac{\partial^2 \mathbf{e}(\theta)}{\partial \theta^2} \right|_{\theta_{\top}} (\hat{\theta} - \theta_{\top})(\hat{\theta} - \theta_{\top})^T \right). \end{aligned} \quad (5)$$

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 [58] thus argue that

$$E[\mathbf{e}(\hat{\theta})] \approx \mathbf{e}_B + (1/2)\text{tr}((\partial^2 \mathbf{e}(\theta)/\partial \theta^2)|_{\theta_T} \text{Cov}(\hat{\theta})),$$

where  $\mathbf{e}_B = \mathbf{e}(\theta_T)$  is the Bayes error rate. They also show that if  $\text{Cov}(\theta') \geq \text{Cov}(\theta'')$  for some  $\theta'$  and  $\theta''$ , then the second term in the approximation is larger for  $\theta'$  than for  $\theta''$ . Because  $\mathbf{I}_u(\theta)$  is always positive definite,  $\mathbf{I}_l(\theta) \leq \mathbf{I}(\theta)$ . Thus, using the Cramer–Rao lower bound (Eq. 4) 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 approaches infinity and is an approximation for finite values.

### 3.2 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  $\mathbf{e}(\theta_u^*) > \mathbf{e}(\theta_l^*)$ .<sup>1</sup> If we observe a large number of labeled samples, the classification error is approximately  $\mathbf{e}(\theta_l^*)$ . If we then collect more samples, most of them unlabeled, we eventually reach a point where the classification error approaches  $\mathbf{e}(\theta_u^*)$ . So the net result is that we started with a classification error close to  $\mathbf{e}(\theta_l^*)$  and, when we added 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  $\lambda$ . 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.

### 3.3 Discussion

Despite the shortcomings of semisupervised learning presented in the previous sections, we do not discourage its use. Understanding the causes of performance degradation with unlabeled data motivates the exploration of new methods attempting to use positively the available unlabeled

<sup>1</sup> We must address 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 [8]. 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.

data. Incorrect modeling assumptions in Bayesian networks culminate mainly as discrepancies in the graph structure, signifying incorrect independence assumptions among variables. To eliminate the increased bias caused by the addition of unlabeled data, we can try simple solutions, such as model switching (Sect. 4.2), or attempt to learn better structures. We describe likelihood-based structure learning methods (Sect. 4.3) and a possible alternative: classification-driven structure learning (Sect. 4.4). In cases where relatively mild changes in structure still suffer from performance degradation from unlabeled data, there are different approaches that can be taken: discard the unlabeled data, give them a different weight (Sect. 4.5), or use the alternative of actively labeling some of the unlabeled data (Sect. 4.6).

To summarize, the main conclusions that can be derived from our analysis are:

- Labeled and unlabeled data contribute to a reduction in variance in semisupervised learning under maximum-likelihood estimation. *This is true regardless of whether or not the model is correct.*
- If 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
- If the model is incorrect, there may be different asymptotic estimation biases for different values of  $\lambda$  (the ratio between the number of labeled and unlabeled data). Asymptotic classification error may also be different for different values of  $\lambda$ . An increase in the number of unlabeled samples may lead to a larger bias from the true distribution and a larger classification error.

In the next section, we discuss several possible solutions for the problem of performance degradation in the framework of Bayesian network classifiers.

## 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 a classifier. If the correct structure is obtained, unlabeled data improve the classifier; otherwise, unlabeled data can actually degrade performance. Somewhat surprisingly, the option of searching for better structures was not proposed by researchers who 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.

### 4.1 Bayesian networks

Bayesian networks [47] are tools for modeling and classification. A Bayesian network (BN) 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 joint probability distribution is factored to the collection of conditional probability distributions of each node in the graph as:

$$p(X_1, \dots, X_n) = \prod_{i=1}^n p(X_i | \Pi_i). \quad (6)$$

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*. In the above notations,  $\mathbf{X}$  is an incoming vector of features. 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  $p(C, \mathbf{X})$  which represents the a posteriori probability of the class given the features.

Maximum-likelihood estimation is one of the main methods to learn the parameters of a network. When there are missing data in a training set, the expectation maximization (EM) algorithm [17] can be used to maximize the likelihood.

As a direct consequence of the analysis in Sect. 3, a BN 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. Therefore, to solve the problem of performance degradation in BNs, we need to take a careful look at the assumed structure of the classifier.

#### 4.2 Switching between simple models

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 [56] in which the features are assumed independent given the class. If performance degrades with unlabeled data, switch to a different type of BN classifier, namely, the Tree-Augmented Naive Bayes classifier (TAN) [26].

In the TAN classifier structure the class node has no parents and each feature has the class node and at most one other feature as parents such that the result is a tree structure for the features. Learning the most likely TAN structure has an efficient and exact solution [26] using a modified Chow–Liu algorithm [11]. Learning the TAN classifiers when there are unlabeled data requires a modification of the original algorithm to what we named the EM-TAN algorithm [14].

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 examples in the experimental section show, switching from NB to TAN does not guarantee that the performance degradation will not occur.

Very relevant is the research of Baluja [1]. The author uses labeled and unlabeled data in a probabilistic classifier framework to detect the orientation of a face. In his study, he obtained excellent classification results, but there were cases where unlabeled data degraded performance. As a consequence, he decided to switch from a Naive Bayes approach to more complex models. Following this intuitive direction we explain Baluja’s observations and provide a solution to the problem: structure learning.

#### 4.3 Beyond simple models

A different approach to overcoming performance degradation is to learn the structure of the BN without restrictions other than the generative one.<sup>2</sup> There are a number of such algorithms in the literature (among them [5, 10, 25]). Nearly all structure-learning algorithms use the ‘likelihood-based’ approach. The goal is to find structures that best fit the data (with perhaps a prior distribution over different structures). Since more complicated structures have higher likelihood scores, penalizing terms are added to avoid overfitting to the data, e.g., the minimum description length (MDL) term. 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 [26]. Therefore, a network with a higher score is not necessarily a better classifier. Friedman et al. [26] suggest changing the scoring function to focus only on the posterior probability of the class variable, but they 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. Therefore, we decided to take a different approach, presented in the next section.

#### 4.4 Classification-driven stochastic structure search

In our approach, instead of trying to estimate the best a posteriori probability, we try to find the structure that minimizes the probability of classification error directly. To do this we designed a classification-driven stochastic search algorithm (SSS) [13, 15]. The basic idea of this approach is that, since we are interested in finding a structure that performs well as a classifier, it would be natural to design an algorithm

<sup>2</sup> A BN classifier is a *generative* classifier when the class variable is an ancestor (e.g., parent) of some (or all) features.

that uses classification error as the guide for structure learning. Here we can further leverage the properties of semisupervised 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 toward finding the optimal classifier.

To learn 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. As we have no simple closed-form expression that relates structure to 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 that we want to maximize:

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

$$\text{inv}_e(S') = \frac{1}{\sum_S \frac{1}{p_S(\hat{c}(\mathbf{X}) \neq C)}}, \quad (7)$$

where the summation is over the space of possible structures and  $p_S(\hat{c}(\mathbf{X}) \neq C)$  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. To construct 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 pre-defined 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,  $S_{\text{new}}$ , to replace a previous structure,  $S_t$ , as follows:

$$\begin{aligned} & \min \left( 1, \left( \frac{\text{inv}_e(S_{\text{new}})}{\text{inv}_e(S_t)} \right)^{1/T} \frac{q(S_t | S_{\text{new}})}{q(S_{\text{new}} | S_t)} \right) \\ & = \min \left( 1, \left( \frac{p_{\text{error}}^t}{p_{\text{error}}^{\text{new}}} \right)^{1/T} \frac{N_t}{N_{\text{new}}} \right), \end{aligned} \quad (8)$$

where  $q(S' | S)$  is the transition probability from  $S$  to  $S'$  and  $N_t$  and  $N_{\text{new}}$  are the sizes of the neighborhoods of  $S_t$  and  $S_{\text{new}}$ , 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 that is aperiodic and irreducible, thus satisfying the Markov chain Monte Carlo (MCMC) conditions [36].

$T$  is used 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 would mostly allow acceptance of structures that improve the 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 1 [29].

The SSS algorithm, with a logarithmic cooling schedule  $T$ , can find a structure that is close to minimum probability of error. We estimate the classification error of a given structure using the labeled training data. Therefore, to avoid overfitting, we add a multiplicative penalty term derived from the Vapnik–Chervonenkis (VC) bound on the empirical classification error. This penalty term penalizes complex classifiers, thus keeping the balance between bias and variance (for more details we refer the reader to [13] and [15]).

#### 4.5 Should unlabeled data be weighed differently?

An interesting strategy, suggested by Nigam et al. [41], is to change the weight of the unlabeled data (reducing their effect on the likelihood). The basic idea in Nigam et al.’s estimators is to produce a modified log-likelihood of the form:

$$\lambda' L_l(\theta) + (1 - \lambda') L_u(\theta), \quad (9)$$

where  $L_l(\theta)$  and  $L_u(\theta)$  are the likelihoods of the labeled and unlabeled data, respectively. For a sequence of  $\lambda'$ , maximize the modified log-likelihood functions to obtain  $\hat{\theta}_{\lambda'}$  ( $\hat{\theta}$  denotes an estimate of  $\theta$ ), and choose the best one with respect to cross-validation or testing. This estimator is simply modifying the ratio of labeled to unlabeled samples for any fixed  $\lambda'$ . Note that this estimator can only make sense under the assumption that the model is incorrect. Otherwise, both terms in Eq. 9) lead to unbiased estimators of  $\theta$ .

Our experiments in [12] suggest that there is then no reason to impose different weights on the data, and much less reason to search for the best weight, when the differences are solely in the rate of reduction of variance. Presumably there are a few labeled samples available and a large number of unlabeled samples; why should we increase the importance of the labeled samples, giving more weight to a term that will contribute more heavily to the variance?

#### 4.6 Active learning

All the methods presented above consider a “passive” use of unlabeled data. A different approach is known as active learning, in which an oracle is queried as to the label of some of the unlabeled data. Such an approach increases the size of the labeled dataset, reduces the classifier’s variance, and thus reduces the classification error. There are different ways to choose which unlabeled data to query. The straightforward approach is to choose a sample randomly.



This approach ensures that the data distribution  $p(C, \mathbf{X})$  is unchanged, a desirable property when estimating generative classifiers. However, the random sample approach typically requires many more samples to achieve the same performance as methods that choose to label data close to the decision boundary. We note that, for generative classifiers, the latter approach changes the data distribution, thereby leading to estimation bias. Nevertheless, McCallum and Nigam [37] used active learning with generative models with success. They proposed to first actively query some of the labeled data followed by estimation of the model’s parameters with the remainder of the unlabeled data.

We performed extensive experiments in [12]. Here we present only the main conclusions. With correctly specified generative models and a large pool of unlabeled data, “passive” use of the unlabeled data is typically sufficient to achieve good performance. Active learning can help reduce the chances of numerical errors (improve EM starting point, for example) and help in the estimation of classification error. With incorrectly specified generative models, active learning is very profitable in quickly reducing the error, while adding the remainder of unlabeled data might not be desirable.

#### 4.7 Summary

The idea of structure search is particularly promising when unlabeled data are present. It seems that simple heuristic methods, such as the solution proposed by Nigam et al. [41] of weighing down the unlabeled data, are not the best strategies for unlabeled data. We suggest that structure search, and in particular stochastic structure search, holds the most promise for handling large amounts of unlabeled data and relatively scarce labeled data for classification. We also believe that the success of structure search methods for classi-

fication increases significantly the breadth of applications of BNs.

In a nutshell, when faced with the option of learning with labeled and unlabeled data, our discussion suggests pursuing the following path. 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, using EM and EM-TAN. If the result is unsatisfactory, then SSS can be used to attempt to further improve performance with enough computational resources. If none of the methods using the unlabeled data improves performance over the supervised TAN (or Naive Bayes), active learning can be used, as long as there are resources to label some samples.

## 5 Experiments

In this section we show our experimental results of BN classifiers learned with labeled and unlabeled data for the three HCI applications discussed in Sect. 2: facial expression recognition, face detection, and skin detection.

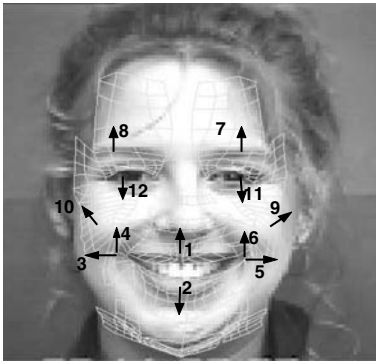
### 5.1 Facial expression recognition experiments

For these experiments we used our real-time facial expression recognition system [14]. This is composed of a face tracking algorithm that outputs a vector of motion features of certain regions of the face. The features are used as inputs to a BN classifier. A snapshot of the system, with the face tracking and the corresponding recognition result, is shown in Fig. 1.

The face tracking we use in our system is based on a system developed by Tao and Huang [59] called the piecewise Bézier volume deformation (PBVD) tracker.



**Fig. 1** A snapshot of our real-time facial expression recognition system. On the right-hand side is a wireframe model overlaid on a face being tracked. On the left-hand side the correct expression, angry, is detected (the bars show the relative probability of angry compared to the other expressions). The subject shown is from the Cohn–Kanade database



**Fig. 2** The facial motion measurements

The face tracker uses a model-based approach where an explicit 3D wireframe model of the face is constructed. In the first frame of the image sequence, landmark facial features such as the eye corners and mouth corners are selected interactively. The generic face model is then warped to fit the selected facial features. The face model consists of 16 surface patches embedded in Bézier volumes. The surface patches defined in this way are guaranteed to be continuous and smooth. The shape of the mesh can be changed by changing the locations of the control points in the Bézier volume.

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 (MUs). Note that they are similar but not equivalent to Ekman’s AUs [21] and are numeric in nature, representing not only the activation of a facial region, but also the direction and intensity of the motion. The 12 MUs used in the face tracker are shown in Fig. 2. The MUs are used as the features for the BN classifiers learned with labeled and unlabeled data.

There are seven categories of facial expressions corresponding to *neutral*, *joy*, *surprise*, *anger*, *disgust*, *sad*, and *fear*. For testing we use two databases, in which all the data are labeled. We remove the labels of most of the training data and learn the classifiers with the different approaches discussed in Sect. 4.

The first database was collected by Chen and Huang [9] and is a database of subjects 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 of the six emotions, starting and ending at the neutral expression. The video sampling rate was 30Hz, and a typical emotion sequence is about 70 samples long ( $\sim 2$  s). The second database is the Cohn–Kanade database [34] and 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, but, because for some of the subjects not all six facial expression sequences were available to us, we used a subset of 53 subjects, for which at least four of the sequences were present. For each subject there is at most one sequence per expression with an average of eight 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). The results are shown in Table 1, showing classification accuracy with 95% confidence intervals. We see that the classifier trained with the SSS algorithm improves classification performance to about 75% for both datasets. Model switching from Naive Bayes to TAN does not significantly improve the performance; apparently, the increase in the likelihood of the data does not cause a decrease in the classification error. In both the NB and TAN cases, we see a performance degradation as the unlabeled data are added to the smaller labeled dataset (TAN-L and NB-L compared to TAN-LUL and NB-LUL). An interesting fact arises from learning the same classifiers with all the data being labeled (i.e., original database without removal of any labels). Now, SSS achieves about 83% accuracy, compared to the 75% achieved with the unlabeled data. Had we had more unlabeled data, it might have been possible to achieve similar performance as with the fully labeled database. This result points to the fact that labeled data are more valuable than unlabeled data (see [8] for a detailed analysis).

## 5.2 Face detection experiments

In our face detection experiments we propose to use BN classifiers, with the image pixels of a predefined window size as the features in the BN. Among the various studies, those of Colmenarez and Huang [16] and Wang et al. [62] are most related to the BN classification methods for face detection. Both learn some “structure” between the facial pixels and combine them to a probabilistic classification

**Table 1** Experimental setup and classification results for facial expression recognition with labeled data (L) and labeled + unlabeled data (LUL). Accuracy is shown with the corresponding 95% confidence interval

Dataset	Train		Test	NB-L	NB-LUL	TAN-L	TAN-LUL	SSS-LUL
	# labeled	# unlabeled						
Chen–Huang	300	11,982	3,555	71.25 $\pm$ 0.75%	58.54 $\pm$ 0.81%	72.45 $\pm$ 0.74%	62.87 $\pm$ 0.79%	74.99 $\pm$ 0.71%
Cohn–Kanade	200	2,980	1,000	72.50 $\pm$ 1.40%	69.10 $\pm$ 1.44%	72.90 $\pm$ 1.39%	69.30 $\pm$ 1.44%	74.80 $\pm$ 1.36%

rule. Both use the entropy between the different pixels to learn pairwise dependencies.

Our approach to detecting faces is an appearance-based one, where the intensity of image pixels serves as the feature for the classifier. In a natural image, faces can appear at different scales, rotations, and location. For learning and defining the BN 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 windows' pixels.

The goal of the classifier is to determine if the pixels in a fixed-size window are those of a face or nonface. While faces are a well-defined concept and have a relatively regular appearance, it is harder to characterize nonfaces. We therefore model the pixel intensities as discrete random variables, as it would be impossible to define a parametric probability distribution function (pdf) for nonface images. For 8-bit representation of pixel intensity, each pixel has 256 values. Clearly, if all these values are used for the classifier, the number of parameters of the joint distribution is too large for learning dependencies among the pixels (as is the case with TAN classifiers). Therefore, there is a need to reduce the number of values representing pixel intensity. Colmenarez and Huang [16] used fourxs values per pixel using fixed and equal bin sizes. We use nonuniform discretization using the class conditional entropy as the mean to bin the 256 values to a smaller number. We use the MLC++ software for that purpose, as described in [19].

Note that our methodology can be extended to other face detection methods that use different features. The complexity of our method is  $O(n)$ , where  $n$  is the number of features (pixels in our case) considered in each image window.

We tested the different approaches described in Sect. 4, with both labeled and unlabeled data. For training the classifier we used a dataset consisting of 2,429 faces and 10,000 nonfaces obtained from the MIT CBCL Face database #1.<sup>3</sup> Examples of face images from the database are presented in Fig. 3. Each face image was cropped and resampled to a  $19 \times 19$  window; thus we had a classifier with 361 features. We also randomly rotated and translated the face images to create a training set of 10,000 face images. In addition, we had available 10,000 nonface images. We left out 1,000 images (faces and nonfaces) for testing and trained the BN classifiers on the remaining 19,000. In all the experiments we learned a Naive Bayes, TAN, and a general generative BN classifier, the latter using the SSS algorithm.

In Table 2 we summarize the results obtained for different algorithms and in the presence of increasing numbers of unlabeled data. We fixed the false alarm to 1, 5, and 10%, and we computed the detection rates. We first learned using all the training data being labeled (that is 19,000 labeled images). The classifier learned with the SSS algorithm outperformed both TAN and NB classifiers, and all performed quite well, achieving high detection rates with a low rate of false alarm. Next we removed the labels of some of the



**Fig. 3** Randomly selected face examples

training data and trained the classifiers. In the first case, we removed the labels of 97.5% of the training data (leaving only 475 labeled images). We see that the NB classifier using both labeled and unlabeled data performed very poorly. The TAN based only on the 475 labeled images and the TAN based on the labeled and unlabeled images were close in performance; thus there was no significant degradation

**Table 2** Detection rates (%) for various numbers of false positives

Detector	False positives		
	1%	5%	10%
<b>NB</b>			
19,000 labeled	74.31	89.21	92.72
475 labeled	68.37	86.55	89.45
475 labeled + 18,525 unlabeled	66.05	85.73	86.98
250 labeled	65.59	84.13	87.67
250 labeled + 18,750 unlabeled	65.15	83.81	86.07
19,000 labeled	91.82	96.42	99.11
475 labeled	86.59	90.84	94.67
475 labeled + 18,525 unlabeled	85.77	90.87	94.21
250 labeled	75.37	87.97	92.56
<b>TAN</b>			
250 labeled + 18,750 unlabeled	77.19	89.08	91.42
19,000 labeled	90.27	98.26	99.87
475 labeled + 18,525 unlabeled	88.66	96.89	98.77
<b>SSS</b>			
250 labeled + 18,750 unlabeled	86.64	95.29	97.93
19,000 labeled	87.78	93.84	94.14
475 labeled	82.61	89.66	91.12
<b>SVM</b>			
250 labeled	77.64	87.17	89.16

<sup>3</sup> CBCL Face Database #1. MIT Center For Biological and Computation Learning, <http://www.ai.mit.edu/projects/cbcl>

of performance after adding the unlabeled data. When only 250 labeled data were used (the labels of about 98.7% of the training data were removed), NB with both labeled and unlabeled data performed poorly, while SSS outperformed the other classifiers with no great reduction of performance compared to the previous cases. For benchmarking, we also implemented a support vector machine classifier (we used the implementation of Osuna et al. [44]). Note that this classifier started off very good but did not improve performance.

In summary, note that the detection rates for NB are lower than those obtained for the other detectors. Overall, the results obtained with SSS are the best. We see that even in the most difficult cases, there was a sufficient amount of unlabeled data to achieve almost the same performance as with a large labeled dataset.

We also tested our system on the CMU test set [52] consisting of 130 images with a total of 507 frontal faces. The results are summarized in Table 3. Note that the results we obtained are comparable to those obtained by Viola and

**Table 3** Detection rates (%) for various numbers of false positives on the CMU test set

Detector	False positives	
	10%	20%
SSS		
19,000 labeled	91.7	92.84
475 labeled + 18,525 unlabeled	89.67	91.03
250 labeled + 18,750 unlabeled	86.64	89.17
Viola-Jones [61]	92.1	93.2
Rowley et al. [52]	–	89.2

Jones [61] and better than those of Rowley et al. [52]. Examples of the detection results on some of the images of the CMU test are presented in Fig. 4. We noticed similar failure modes as Viola and Jones [61]. Since the face detector was trained only on frontal faces, our system failed to detect faces if they had a significant rotation out of the plane (toward a profile view). The detector also had problems with



**Fig. 4** Output of the system on some images of the CMU test using the SSS classifier learned with 19,000 labeled data. *MFs* represents the number of missed faces, and *FDs* is the number of false detections





Fig. 5 Examples of detected skin patches

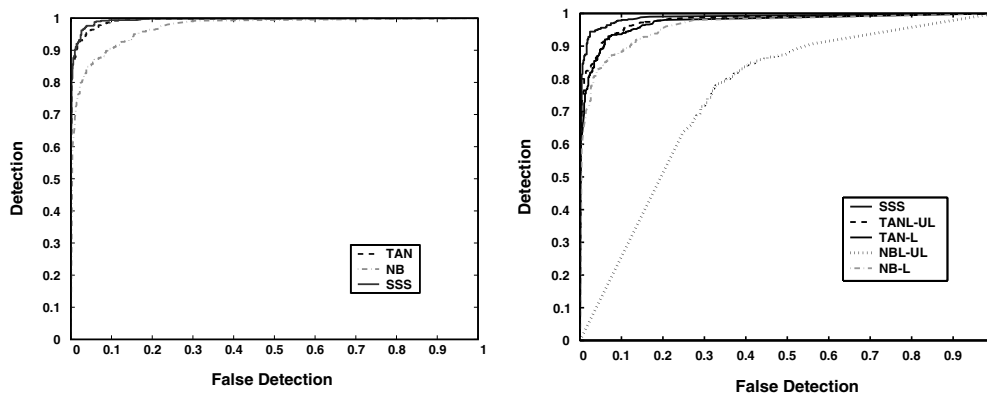


Fig. 6 ROC curves showing detection rates of skin compared to false detection with all data labeled (*left*) and 90% unlabeled data (*right*): SSS, NB learned with labeled data only (NB-L) and with labeled and unlabeled data (NB-LUL), and TAN learned with labeled data only (TAN-L) and with labeled and unlabeled data (TAN-LUL)

the images in which the faces appeared dark and the background was relatively light. Inevitably, we also detected false positives especially in some texture regions.

### 5.3 Skin detection experiments

In our experiments, we used image patches of nine pixels (a  $3 \times 3$  patch) as the features in the Bayesian network. We considered the  $rg$  chromaticity space, which is the most popular color space for skin color modeling [66].

We used the database of Jones and Rehg [33], which consists of 3,475 images containing skin and 8,796 non-skin images. Each image was manually segmented such that the skin pixels were labeled. Examples of detected skin patches are presented in Fig. 5. In the experiments, we randomly selected  $3 \times 3$  skin and nonskin patches (100,000 in total). We left out 40,000 patches for testing and trained the BN classifiers on the remaining 60,000. To compare the results of the classifiers, we used the receiving operating characteristic (ROC) curves. The ROC curves show, under different classification thresholds ranging from 0 to 1, the probability of detecting a skin patch in a skin image,  $P_D = P(\hat{C} = \text{skin} | C = \text{skin})$ , against the probability of falsely detecting a skin patch in a nonskin image,  $P_{FD} = P(\hat{C} = \text{nonskin} | C \neq \text{nonskin})$ .

We first learned using all the training data being labeled (that is, 60,000 labeled patches). Figure 6 (left) shows the resultant ROC curve for this case. The classifier learned with the SSS algorithm outperformed both TAN and NB classifiers, and all performed quite well, achieving high detection rates with a low rate of false alarm. Next we removed the labels of some of the training data and trained the classifiers. Figure 6 (right) shows the case where the labels of 90% of the training data (leaving only 600 labeled patches) were removed. We see that the NB classifier using both labeled and unlabeled data (NB-LUL) performed very poorly. The TAN based only on the 600 labeled images (TAN-L) and the TAN based on the labeled and unlabeled images (TAN-LUL) were close in performance, and thus there was no significant degradation of performance when adding the unlabeled data.

## 6 Conclusion

In this work we presented a Bayesian network approach for three human–computer interaction applications: facial expression recognition, face detection, and skin detection. We considered several instances of Bayesian networks and showed that learning the structure of Bayesian network classifiers enables learning good classifiers with a small labeled set and a large unlabeled set.

Our discussion of semisupervised 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 improves performance over the supervised TAN (or Naive Bayes), either discard the unlabeled data or try to label more data, using active learning, for example.

In closing, it is possible to view some of the components of this work independently of each other. The theoretical results of Sect. 3 do not depend on the choice of probabilistic classifier and can be used as a guide to other classifiers. Structure learning of Bayesian networks is not a topic motivated solely by the use of unlabeled data. The three applications we considered could be solved using classifiers other than Bayesian networks. However, this work should be viewed as a combination of all three components: (1) the theory showing the limitations of unlabeled data is used to motivate (2) the design of algorithms to search for better-performing structures of Bayesian networks, and, finally, (3) the successful applications to a human-computer interaction problem we are interested in solving by learning with labeled and unlabeled data.

## References

- Baluja, S.: Probabilistic modelling for face orientation discrimination: learning from labeled and unlabeled data. In: *Neural Information and Processing Systems*, pp. 854–860 (1998)
- Black, M.J., Yacoob, Y.: Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In: *Proceedings of the International Conference on Computer Vision*, pp. 374–381 (1995)
- Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: *Conference on Learning Theory*, pp. 92–100 (1998)
- Bosson, A., Cawley, G., Chan, Y., Harvey, R.: Non-retrieval: blocking pornographic images. In: *International Conference on Image and Video Retrieval*, pp. 50–60 (2002)
- Brand, M.: An entropic estimator for structure discovery. In: *Neural Information and Processing Systems*, pp. 723–729 (1998)
- Bruce, R.: Semi-supervised learning using prior probabilities and EM. In: *International Joint Conference on Artificial Intelligence, Workshop on Text Learning: Beyond Supervision* (2001)
- Caetano, T., Olabarriaga, S., Barone, D.: Do mixture models in chromaticity space improve skin detection? *Pattern Recog.* **36**, 3019–3021 (2003)
- Castelli, V.: The relative value of labeled and unlabeled samples in pattern recognition. PhD Thesis, Stanford University, Stanford, CA (1994)
- Chen, L.S.: Joint processing of audio-visual information for the recognition of emotional expressions in human-computer interaction. PhD Thesis, University of Illinois at Urbana-Champaign (2000)
- Cheng, J., Greiner, R., Kelly, J., Bell, D.A., Liu, W.: Learning Bayesian networks from data: an information-theory based approach. *Artif. Intell. J.* **137**, 43–90 (2002)
- Chow, C.K., Liu, C.N.: Approximating discrete probability distribution with dependence trees. *IEEE Trans. Inf. Theory* **14**, 462–467 (1968)
- Cohen, I.: Semi-supervised learning of classifiers with application to human computer interaction. PhD Thesis, University of Illinois at Urbana-Champaign (2003)
- Cohen, I., Sebe, N., Cozman, F., Cirelo, M., Huang, T.S.: Learning Bayesian network classifiers for facial expression recognition using both labeled and unlabeled data. In: *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 595–601 (2003)
- Cohen, I., Sebe, N., Garg, A., Chen, L., Huang, T.S.: Facial expression recognition from video sequences: temporal and static modelling. *Comput. Vis. Image Understand.* **91**(1–2), 160–187 (2003)
- Cohen, I., Cozman, F., Sebe, N., Cirello, M., Huang, T.S.: Semi-supervised learning of classifiers: theory, algorithms, and their applications to human-computer interaction. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(12), 1553–1567 (2004)
- Colmenarez, A.J., Huang, T.S.: Face detection with information based maximum discrimination. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 782–787 (1997)
- Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B* **39**(1), 1–38 (1977)
- Donato, G., Bartlett, M.S., Hager, J.C., Ekman, P., Sejnowski, T.J.: Classifying facial actions. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(10), 974–989 (1999)
- Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: *International Conference on Machine Learning*, pp. 194–202 (1995)
- Ekman, P.: Strong evidence for universals in facial expressions: a reply to Russell’s mistaken critique. *Psychol. Bull.* **115**(2), 268–287 (1994)
- Ekman, P., Friesen, W.V.: *Facial Action Coding System: Investigator’s Guide*. Consulting Psychologists Press, Palo Alto, CA (1978)
- Essa, I.A., Pentland, A.P.: Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(7), 757–763 (1997)
- Fasel, B., Luetttin, J.: Automatic facial expression analysis: a survey. *Pattern Recog.* **36**, 259–275 (2003)
- Fleck, M., Forsyth, D., Bregler, C.: Finding naked people. In: *European Conference on Computer Vision*, pp. 593–602 (1996)
- Friedman, N.: The Bayesian structural EM algorithm. In: *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 129–138 (1998)
- Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. *Mach. Learn.* **29**(2), 131–163 (1997)
- Ghani, R.: Combining labeled and unlabeled data for multi-class text categorization. In: *International Conference on Machine Learning*, pp. 187–194 (2002)
- Goleman, D.: *Emotional Intelligence*. Bantam Books, New York (1995)
- Hajek, B.: Cooling schedules for optimal annealing. *Math. Oper. Res.* **13**, 311–329 (1988)
- Heisele, B., Ho, P., Wu, J., Poggio, T.: Face recognition: component-based versus global approaches. *Comput. Vis. Image Understand.* **91**(1–2), 6–21 (2003)
- Hjelmas, E., Low, B.K.: Face detection: a survey. *Comput. Vis. Image Understand.* **83**, 236–274 (2003)
- Jedynak, B., Zheng, H., Daoudi, M.: Statistical models for skin detection. In: *IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Statistical Analysis in Computer Vision* (2003)
- Jones, M.J., Rehg, J.M.: Statistical color models with application to skin detection. *Int. J. Comput. Vis.* **46**(1), 81–96 (2002)
- Kanade, T., Cohn, J.F., Tian, Y.: Comprehensive database for facial expression analysis. In: *International Conference on Automatic Face and Gesture Recognition*, pp. 46–53 (2000)

35. Kouzani, A.Z.: Locating human faces within images. *Comput. Vis. Image Understand.* **91**(3), 247–279 (2003)
36. Madigan, D., York, J.: Bayesian graphical models for discrete data. *Int. Stat. Rev.* **63**, 215–232 (1995)
37. McCallum, A.K., Nigam, K.: Employing EM in pool-based active learning for text classification. In: *International Conference on Machine Learning*, pp. 350–358 (1998)
38. McKenna, S.J., Gong, S., Raja, Y.: Modeling facial colour and identity with Gaussian mixtures. *Pattern Recog.* **31**, 1883–1892 (1998)
39. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculation by fast computing machines. *J. Chem. Phys.* **21**, 1087–1092 (1953)
40. Nefian, A., Hayes, M.: Face recognition using an embedded HMM. In: *IEEE Conference on Audio and Video-Based Biometric Person Authentication*, pp. 19–24 (1999)
41. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. *Mach. Learn.* **39**, 103–134 (2000)
42. Oliver, N., Pentland, A., Bérard, F.: LAFTER: a real-time face and lips tracker with facial expression recognition. *Pattern Recog.* **33**, 1369–1382 (2000)
43. O’Neill, T.J.: Normal discrimination with unclassified observations. *J. Am. Stat. Assoc.* **73**(364), 821–826 (1978)
44. Osuna, E., Freund, R., Girosi, F.: Training support vector machines: an application to face detection. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 130–136 (1997)
45. Pantic, M., Rothkrantz, L.J.M.: Automatic analysis of facial expressions: the state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(12), 1424–1445 (2000)
46. Pantic, M., Rothkrantz, L.J.M.: Toward an affect-sensitive multimodal human-computer interaction. *Proc. IEEE* **91**(9), 1370–1390 (2003)
47. Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA (1988)
48. Pentland, A.: Looking at people. *Commun. ACM* **43**(3), 35–44 (2000)
49. Pentland, A., Choudhury, T.: Face recognition for smart environments. *IEEE Comput.* **33**(2), 50–55 (2000)
50. Ratsaby, J., Venkatesh, S.S.: Learning from a mixture of labeled and unlabeled examples with parametric side information. In: *Conference on Computational Learning Theory*, pp. 412–417 (1995)
51. Rosenblum, M., Yacoob, Y., Davis, L.S.: Human expression recognition from motion using a radial basis function network architecture. *IEEE Trans. Neural Netw.* **7**(5), 1121–1138 (1996)
52. Rowley, H., Baluja, S., Kanade, T.: Neural network-based face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(1), 23–38 (1998)
53. Saxe, D., Foulds, R.: Toward robust skin identification in video images. In: *Automatic Face and Gesture Recognition*, pp. 379–384 (1996)
54. Schneiderman, H.: Learning a restricted Bayesian network for object detection. In: *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 639–646 (2004)
55. Schwerdt, K., Crowley, J.L.: Robust face tracking using color. In: *Automatic Face and Gesture Recognition*, pp. 90–95 (2000)
56. Sebe, N., Cohen, I., Garg, A., Lew, M.S., Huang, T.S.: Emotion recognition using a Cauchy naive Bayes classifier. In: *International Conference on Pattern Recognition* (2002)
57. Seeger, M.: *Learning with labeled and unlabeled data*. Technical Report, Edinburgh University, Edinburgh, UK (2001)
58. Shahshahani, B., Landgrebe, D.: Effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Trans. Geosci. Remote Sens.* **32**(5), 1087–1095 (1994)
59. Tao, H., Huang, T.S.: Connected vibrations: a modal analysis approach to non-rigid motion tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 735–740 (1998)
60. Terrillon, J.-C., Shirazi, M.N., Fukamachi, H., Akamatsu, S.: Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In: *Automatic Face and Gesture Recognition*, pp. 54–61 (2000)
61. Viola, P., Jones, M.J.: Robust real-time object detection. *Int. J. Comput. Vis.* **57**(2) (2004)
62. Wang, R.R., Huang, T.S., Zhong, J.: Generative and discriminative face modeling for detection. In: *Automatic Face and Gesture Recognition* (2002)
63. White, H.: Maximum likelihood estimation of misspecified models. *Econometrica* **50**(1), 1–25 (1982)
64. Yang, M.-H., Ahuja, N.: Detecting human faces in color images. In: *International Conference on Image Processing*, pp. 127–130 (1998)
65. Yang, M.-H., Roth, D., Ahuja, N.: SNoW based face detector. In: *Neural Information Processing Systems*, pp. 855–861 (2000)
66. Yang, M.-H., Kriegman, D., Ahuja, N.: Detecting faces in images: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(1), 34–58 (2002)
67. Zhang, T., Oles, F.: A probability analysis on the value of unlabeled data for classification problems. In: *International Conference on Machine Learning* (2000)