

Human-Computer Interaction: A Bayesian Network Approach

Nicu Sebe*, Ira Cohen[†], Thomas S. Huang[‡] and Theo Gevers*

*Faculty of Science, University of Amsterdam, The Netherlands

[†] HP Research Labs, USA

[‡] Beckman Institute, University of Illinois at Urbana-Champaign, USA

Abstract—Human-computer interaction (HCI) lies at the crossroads of many scientific areas including artificial intelligence, computer vision, face recognition, motion tracking, etc. In this paper, we discuss training probabilistic classifiers with labeled and unlabeled data for human-computer interaction applications. We provide an analysis which 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.

I. INTRODUCTION

Information systems are ubiquitous in all human endeavors including scientific, medical, military, transportation, and consumer. Individual users use them for learning, searching for information, doing research, and authoring. Multiple 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 that we are addressing in this paper. For doing this, in the following, we discuss training probabilistic classifiers for three essential components of such an information system for human-computer interaction (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 [1] and person tracking [2], to pornography filtering [3]. We are especially interested in skin detection as a cue for detecting people (and their faces) in real-world photographs and live videos. 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 within the scene may cast shadows or reflect additional light and so forth. Finally, there are many other objects which are easily confused with skin: certain types of wood, copper, sand as well as clothes often have skin-like colors.

Many of the recent applications designed for HCI 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 [4]. 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 non-rigid object which can change its appearance. The second set of reasons are environmental: changing lighting, rotations, translations, and scales of the faces in natural images. There have been numerous approaches for face detection and we refer the reader to the reviews of Yang et al. [1] and Hjelmas and Low [5].

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. 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. Ekman and Friesen [6] 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) where each AU has some related muscular basis. 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 recent surveys in the area [7], [8], [9] provide an in depth review of many of the research done in automatic facial expression recognition in recent years.

In this paper we are interested in two aspects. First, most of the previous research tried to classify each observable independent from 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 use of labeled and unlabeled data (also called semi-supervised learning). The motivation for semi-supervised 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 label 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. A discussion on how to incorporate unlabeled data in learning Bayesian Networks and the corresponding effect on the classification results is presented in Section III. The experimental analysis is presented in Section IV.

II. BAYESIAN NETWORKS (BN) CLASSIFIERS

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

We consider probabilistic classifiers that represent the a-posteriori probability of the class given the features, $p(C, \mathbf{X})$, using Bayesian networks. A Bayesian network is composed of a directed acyclic graph in which every node is associated with a variable X_i and with a conditional distribution $p(X_i|\Pi_i)$, where Π_i denotes the parents of X_i in the graph. The 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.

If the features in \mathbf{X} are assumed to be independent of each other conditioned upon the class label c we have the Naive Bayes framework [10]. Friedman, et al. [11] proposed the use of the Tree-Augmented Naive Bayes (TAN) model as a classifier, to enhance the performance over the simple NB classifier. 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. For learning the TAN classifier we do not fix the structure of the Bayesian network, but we try to find the TAN structure that maximizes the likelihood function given the training data out of all possible TAN structures [10].

III. SEMI-SUPERVISED LEARNING OF BN

Is there value to 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. Therefore, semi-supervised learning is seen optimistically as a learning paradigm that can relieve the practitioner from the need to collect many expensive labeled training data.

Consider the following scenario. A sample (c, \mathbf{x}) is generated from $p(C, \mathbf{X})$. The value c is then either revealed, and the

sample is a *labeled* one; or the value c is hidden, and the sample is an *unlabeled* one. The probability that any sample is labeled, denoted by λ , is fixed, known, and independent of the samples. Thus the same underlying distribution $p(C, \mathbf{X})$ models both labeled and unlabeled data. Given a set of N_l labeled samples and N_u unlabeled samples, we use maximum likelihood for estimating $\hat{\theta}$. The assumed distribution $p(C, \mathbf{X}|\theta)$ can be decomposed either as $p(C|\mathbf{X}, \theta) p(\mathbf{X}|\theta)$ or as $p(\mathbf{X}|C, \theta) p(C|\theta)$. A parametric model where both $p(\mathbf{X}|C, \theta)$ and $p(C|\theta)$ depend explicitly on θ is referred to as a *generative model*. The log-likelihood function of this model for a dataset with labeled and unlabeled data is:

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

$$L_l(\theta) = \sum_{i=1}^{N_l} \log \left[\prod_C (p(C = c'|\theta) p(\mathbf{x}_i|c', \theta))^{I_{\{C=c'\}}(c_i)} \right]$$

$$L_u(\theta) = \sum_{j=(N_l+1)}^{N_l+N_u} \log \left[\sum_C p(C = c'|\theta) p(\mathbf{x}_j|c', \theta) \right] = \sum_{j=(N_l+1)}^{N_l+N_u} \log [p(\mathbf{x}_j|\theta)]$$

where $I_A(Z)$ is the indicator function (1 if $Z \in A$; 0 otherwise) and $p(C = c')$ are the mixing coefficients. $L_l(\theta)$ and $L_u(\theta)$ are the likelihoods of the labeled and unlabeled data, respectively. 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 [12].

Despite the optimistic view mentioned above, 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, in contrast to improvement of performance when adding more labeled data. In [12] we present an extensive analysis 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 considered a classification driven stochastic structure search (SSS) algorithm for learning the structure of Bayesian network classifiers that minimizes the probability of classification error (see [12] for more details). The advantages of the SSS algorithm are that it usually converges to better classifiers compared to other methods, and asymptotically can be shown to converge to the classifier with minimum error. Its biggest disadvantage is in the added complexity: for every structure being tested the parameters are estimated, followed by error estimation.

IV. EXPERIMENTS

A. Facial Expression Recognition Experiments

For these experiments we used our real time facial expression recognition system [10]. This 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. There are seven categories of facial expressions corresponding to *neutral, joy, surprise, anger, disgust,*

TABLE I

THE EXPERIMENTAL SETUP AND THE 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±0.75%	58.54±0.81%	72.45±0.74%	62.87±0.79%	74.99±0.71%
Cohn-Kanade	200	2,980	1,000	72.50±1.40%	69.10±1.44%	72.90±1.39%	69.30±1.44%	74.80±1.36%

sad, and *fear*. For testing we use two databases, in which all the data is labeled. We removed the labels of most of the training data and learned the classifiers with the different approaches discussed previously.

The first database was collected by Chen and Huang [10] and is a database of subjects that were instructed to display facial expressions corresponding to the six types of emotions. All the tests of the algorithms are performed on a set of five people, each one displaying six sequences of each one of the six emotions, starting and ending at the Neutral expression. The second database is the Cohn-Kanade database [10] 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 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 present.

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 I, 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., the 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 [13] for a detailed analysis).

B. 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 different works, those of Colmenarez [14] and Wang et al. [15] are more related to the Bayesian network classification methods for face detection.

Our approach in detecting faces is an appearance based approach, where the intensity of image pixels serve as the features for the classifier. In a natural image, faces can appear

at different scales, rotations, and location. For learning and defining the Bayesian network classifiers, we must look at fixed size windows and learn how a face appears in such windows, where we assume that the face appears in most of the window's pixels. The goal of the classifier is to determine if the pixels in a fixed size window are those of a face or a non-face. While faces are a well defined concept, and have a relatively regular appearance, it is harder to characterize non-faces. We therefore model the pixel intensities as discrete random variables, as it would be impossible to define a parametric probability distribution function (pdf) for non-face 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 between the pixels (as is the case of TAN classifiers). Therefore, we use non-uniform discretization using the class conditional entropy as the mean to bin the 256 values to a smaller number.

We test the different approaches with both labeled and unlabeled data. For training the classifier we used a dataset consisting of 2,429 faces and 10,000 non-faces obtained from the MIT CBCL Face database. Each face image is cropped and resampled to a 19×19 window, thus we have a classifier with 361 features. We also randomly rotate and translate the face images to create a training set of 10,000 face images. In addition we have available 10,000 non-face images. We leave out 1,000 images (faces and non-faces) for testing and train the Bayesian network classifiers on the remaining 19,000.

TABLE II
DETECTION RATES (%) FOR DIFFERENT FALSE POSITIVES

Detector		False detections		
		1%	5%	10%
NB	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
TAN	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
SSS	19,000 labeled	90.27	98.26	99.87
	475 labeled + 18,525 unlabeled	88.66	96.89	98.77

In Table II we summarize the results obtained for different algorithms and in the presence of increasing number of unlabeled data. We fixed the false alarm to 1%, 5%, and 10% and we computed the detection rates. We first learn using all the training data being labeled (that is 19,000 labeled images). The classifier learned with the SSS algorithm outperforms both TAN and NB classifiers, and all perform quite well, achieving high detection rates with a low rate of false alarm. Next, we remove 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 performs very poorly. The TAN based only on the 475 labeled images and the TAN based on the labeled and unlabeled images are close in performance, thus there was no significant degradation of performance when adding the unlabeled data. The SSS outperforms the other classifiers.

C. Skin Detection Experiments

In our experiments we use image patches of 9 pixels (a 3x3 patch) as the features in the Bayesian Network. We consider the *rg* chromaticity space, which is the most popular color space for skin color modeling [1].

We use the database of Jones and Rehg [16] consists of 3,475 images containing skin and 8,796 non-skin images. Each image was manually segmented such that the skin pixels are labeled. In the experiments we randomly selected 3x3 skin and non-skin patches (100,000 in total). We leave out 40,000 patches for testing and train the Bayesian Network classifiers on the remaining 60,000. To compare the results of the classifiers, we use the receiving operating characteristic (ROC) curves. The ROC curves show, under different classification thresholds, ranging from 0 to 1, the probability of detecting a skin patch in a skin image, $P_D = P(\hat{C} = skin | C = skin)$, against the probability of falsely detecting a skin patch in a non-skin image, $P_{FD} = P(\hat{C} = non - skin | C \neq non - skin)$.

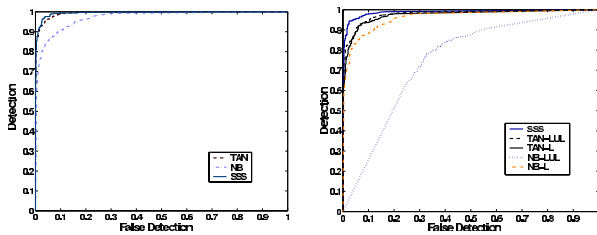


Fig. 1. ROC curves showing detection rates of skin compared to false detection with all data labeled (left) and 90% unlabeled data (right).

We first learn using all the training data being labeled (that is 60,000 labeled patches). Figure 1 (left) shows the resultant ROC curve for this case. The classifier learned with the SSS algorithm outperforms both TAN and NB classifiers, and all perform quite well, achieving high detection rates with a low rate of false alarm. Next we remove the labels of some of the training data and train the classifiers. Figure 1 (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) performs 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) are close in performance, thus there was no significant degradation of performance when adding the unlabeled data. Overall, the results obtained with SSS are the best. We see that even in the most difficult cases, there was sufficient amount of unlabeled data to achieve almost the same performance as with a large sized labeled dataset.

V. SUMMARY AND DISCUSSION

In this work we presented a Bayesian Network approach for three HCI applications. We considered several instances

of Bayesian Networks and we suggested a methodology to perform skin detection using both labeled and unlabeled data.

In a nutshell, when faced with the option of learning with labeled and unlabeled data using Bayesian networks, our discussion suggests using the following path. Start with Naive Bayes and TAN classifiers, learn only with the available 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. If none of the methods using the unlabeled data improve performance over the supervised TAN (or Naive Bayes) the practitioner is faced with two options: discard the unlabeled data, or label some of the unlabeled data using the active learning methodology. Of course, active learning can be used as long as there are resources to label some samples.

Structure learning of Bayesian networks is not a topic motivated solely by the use of unlabeled data. Skin detection could be solved using classifiers other than Bayesian networks. However, this work should be viewed as a combination of 3 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 application by learning with labeled and unlabeled data.

REFERENCES

- [1] M.-H. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *PAMI*, vol. 24, no. 1, pp. 34–58, 2002.
- [2] K. Scherdt and J. Crowley, "Robust face tracking using color," in *FG*, 2000, pp. 90–95.
- [3] M. Fleck, D. Forsyth, and C. Bregler, "Finding naked people," in *ECCV*, 1996, pp. 593–602.
- [4] A. Pentland, "Looking at people," *Communications of the ACM*, vol. 43, no. 3, pp. 35–44, 2000.
- [5] E. Hjelmas and B. Low, "Face detection: A survey," *CVIU*, vol. 83, pp. 236–274, 2003.
- [6] P. Ekman and W. Friesen, *Facial Action Coding System: Investigator's Guide*. Consulting Psychologists Press, 1978.
- [7] M. Pantic and L. Rothkrantz, "Automatic analysis of facial expressions: The state of the art," *PAMI*, vol. 22, no. 12, pp. 1424–1445, 2000. [Online]. Available: citeseer.nj.nec.com/pantic00automatic.html
- [8] B. Fasel and J. Luetttin, "Automatic facial expression analysis: A survey," *Pattern Recognition*, vol. 36, pp. 259–275, 2003.
- [9] N. Sebe, I. Cohen, and T. Huang, "Multimodal emotion recognition," in *Handbook of Pattern Recognition and Computer Vision*, 2005, pp. 387–409.
- [10] I. Cohen, N. Sebe, A. Garg, L. Chen, and T. Huang, "Facial expression recognition from video sequences: Temporal and static modeling," *CVIU*, vol. 91, no. 1–2, pp. 160–167, 2003.
- [11] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, no. 2, pp. 131–163, 1997.
- [12] I. Cohen, F. Cozman, N. Sebe, M. Cirello, and T. Huang, "Semi-supervised learning of classifiers: Theory and algorithms and their applications to human-computer interaction," *PAMI*, vol. 26, no. 12, pp. 1553–1567, 2004.
- [13] V. Castelli, "The relative value of labeled and unlabeled samples in pattern recognition," Ph.D. dissertation, Stanford, 1994.
- [14] A. Colmenarez and T. Huang, "Face detection with information based maximum discrimination," in *CVPR*, 1997, pp. 782–787.
- [15] R. Wang, T. Huang, and J. Zhong, "Generative and discriminative face modeling for detection," in *FG*, 2002.
- [16] M. Jones and J. Rehg, "Statistical color models with application to skin detection," *IJCV*, vol. 46, no. 1, pp. 81–96, 2002.