Semi-Supervised Learning for Facial Expression Recognition

Ira Cohen[1], Nicu Sebe[2], Fabio G. Cozman[3],Thomas S. Huang[4]

[1]HP Labs, Palo Alto, CA, USA
Ira.cohen@hp.com

[2]Faculty of Science, University of Amsterdam, The Netherlands
nicu@science.uva.nl

[3]Escola Politécnica, Universidade de São Paulo, Brazil
fgcozman@usp.br

[4]Beckman Institute, University of Illinois at Urbana-Champaign, IL, USA
huang@ifp.uiuc.edu

ABSTRACT

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

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*learning*

General Terms

Algorithms

Keywords

Semi-supervised learning, Bayesian networks, facial expression recognition

MIR'03, November 7, 2003, Berkeley, California, USA.

1. INTRODUCTION

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 work for humans 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 four-year life span.

It is evident from the above story that it is not sufficient for a machine (computer) to look like a human. Something else is essential: the ability to acquire the emotions. Moreover, the machine should learn to recognize and understand these emotions to be able to have a human-like interaction with its human counterpart. It is argued that to truly achieve effective human-computer intelligent interaction (HCII), there is a need for the computer to be able to interact naturally with the user, similar to the way human-human interaction takes place. 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. 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 do a great deal of inference 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. There is a growing amount of evidence showing that emotional skills are part of what is called "intelligence" [29,

19]. Computers today, on the other hand, are still quite "emotionally challenged." They neither recognize the user's emotions nor possess emotions of their own.

The most expressive way humans display emotions is through facial expressions. Humans detect and interpret faces and facial expressions in a scene with little or no effort. Still, development of 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 a human-like interaction between the man and machine.

This paper tries to make a small dent in the huge task of providing computers with the ability to understand humans. The focus of the paper is on the task of learning how to classify events from data which is labeled and unlabeled. The theoretical and algorithmic results are then applied to facial expression recognition.

2. SEMI-SUPERVISED LEARNING

An important component of image understanding is the ability to classify objects, sequences, or events to different characterizing classes. Learning classifiers is typically done with training data, which can be either labeled to the different classes, or unlabeled. Learning with labeled data is known as supervised learning. This paper is concerned with the use of unlabeled data in supervised learning of classifiers, i.e., a set of labeled data is appended with a, typically much larger, set of unlabeled data. This learning paradigm is known as 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, e.g., in object classification, unlabeled data are all the images in a database, while labeled data require the manual labeling of each image to one of the object classes.

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. Due to an increasing number of applications and algorithms that successfully use unlabeled data [3, 25, 32, 1, 5, 2, 18, 31] and magnified by theoretical issues over the value of unlabeled data in certain cases [6, 28, 26], 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. 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 [25, 32, 1, 5], in contrast to improvement of performance when adding more labeled data. Intrigued by these discrepancies, we performed extensive experiments, reported in [10]. 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.

Despite the shortcomings of semi-supervised learning, 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 data. We restrict ourselves mainly to classifiers based on Bayesian networks (Section 3.1). 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 (Section 3.2) or attempt to learn better structures. We describe likelihood based structure learning methods (Section 3.3) and a possible alternative: classification driven structure learning (Section 3.4). 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 or give them a different weight (Section 3.5), or use the alternative of actively labeling some of the unlabeled data (Section 3.6).

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

- Labeled and unlabeled data contribute to a reduction in variance in semi-supervised learning under maximum likelihood estimation. *This is true regardless of whether the model is correct or not.*
- 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 λ (the ratio between the number of labeled and unlabeled data). Asymptotic classification error may also be different for different values of λ. An increase in the number of unlabeled samples may lead to a larger 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.

3. 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 that previously witnessed the performance degradation. Apparently, performance degradation was attributed to unpredictable, stochastic disturbances in modeling assumptions, and not to mistakes in the underlying structure – something that can be detected and fixed.

3.1 Bayesian Networks

Bayesian networks [27] 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 Π_i denotes the parents of X_i in the graph. The joint probability distribution is factored to the collection of conditional probability distributions of each node in the graph as:

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

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 (MU's - see Section 4). Each instantiation of $\mathbf{X}$ is a *record*. We assume that there is a *class variable* C; the values of C are the *labels*, one of the facial expressions. The classifier receives a record $\mathbf{x}$ and generates a label $\hat{c}(\mathbf{x})$. An optimal classification rule can be obtained from the exact distribution $p(C, \mathbf{X})$ 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 the network. When there are missing data in training set, the Expectation Maximization (EM) algorithm [13] can be used to maximize the likelihood.

As a direct consequence of the analysis in Section 2, a Bayesian network that has the correct structure and the correct parameters is also optimal for classification because the a-posteriori distribution of the class variable is accurately represented. Therefore, to solve the problem of performance degradation in BNs, we need to take a careful look at the assumed structure of the classifier.

3.2 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 [30] in which the features are assumed independent given the class. If performance degrades with unlabeled data, switch to a different type of Bayesian network classifier, namely the Tree-Augmented Naive Bayes classifier (TAN) [15].

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 [17] using a modified Chow-Liu algorithm [9]. Learning the TAN classifiers when there are unlabeled data requires a modification of the original algorithm to what we named the EM-TAN algorithm [12].

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.

3.3 Beyond Simple Models

A different approach to overcome performance degradation is to learn the structure of the Bayesian network without restrictions other than the generative one[1]. There are a number of such algorithms in the literature (among them [21, 16, 4, 8]). 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 overfiting 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 [17, 20]. Therefore, a network with a higher score is not necessarily a better classifier. Friedman et al [17] suggest changing the scoring function to focus only on the posterior probability of the class variable, but show that it is not computationally feasible.

The drawbacks of likelihood based structure learning algorithms could be magnified when learning with unlabeled data; the posterior probability of the class has a smaller effect during the search, while the marginal of the features would dominate. Therefore, we decided to take a different approach presented in the next section.

3.4 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 so we designed a classification driven stochastic search algorithm (SSS) [11].

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

DEFINITION 1. *The* inverse error measure *for structure S' is*

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

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 [24] to generate samples from the inverse error measure, without having to ever compute it for all possible structures. 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 [11]).

3.5 Should Unlabeled Be Weighed Differently?

An interesting strategy, suggested by Nigam et al [25] 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 that is of the form:

$$\lambda' L_l(\theta) + (1 - \lambda')L_u(\theta) \quad (3)$$

where $L_l(\theta)$ and $L_u(\theta)$ are the likelihoods of the labeled and unlabeled data, respectively. For a sequence of λ', maximize the modified log-likelihood functions to obtain $\hat{\theta}_{\lambda'}$ ($\hat{\theta}$ denotes an estimate of θ), 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 λ'. Note that this estimator can only make sense under the assumption that the model is incorrect.

[1] A Bayesian network classifier is a *generative* classifier when the class variable is an ancestor (e.g., parent) of some (or all) features.

Figure 1: A snap shot of our realtime facial expression recognition system. On the right side is a wireframe model overlayed on a face being tracked. On the left side the correct expression, Angry, is detected (the bars show the relative probability of Angry compared to the other expressions). The subject shown is from the Cohn-Kanade database.

Otherwise, both terms in Expression (3) lead to unbiased estimators of θ.

Our experiments in [10] 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 them more weight to a term that will contribute more heavily to the variance?

3.6 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 data set, 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 therefore leading to estimation bias. Nevertheless, McCallum and Nigam [23] 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 [10]. 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 error, while adding the remainder of unlabeled data might not be desirable.

3.7 Concluding Remarks

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 [25] 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 amount of unlabeled data and relatively scarce labeled data for classification. We also believe that the success of structure search methods for classification increases significantly the breadth of applications of Bayesian networks.

In a nutshell, when faced with the option of learning with labeled and unlabeled data, our discussion suggests following 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 not satisfactory, then SSS can be used to attempt to further improve performance with enough computational resources. If none of the methods using the unlabeled data improve performance over the supervised TAN (or Naive Bayes), active learning can be used, as long as there are resources to label some samples.

4. FACIAL EXPRESSION RECOGNITION SYSTEM

Our real time facial expression recognition system is composed of a face tracking algorithm which outputs a vector of motion features of certain regions of the face. The features are used as inputs to a Bayesian network classifier. A snap shot of the system, with the face tracking and the corresponding recognition result is shown in Figure 1.

Figure 3: Examples of images from the video sequences used in the experiment. Top row shows subjects from the Chen-Huang DB, bottom row shows subjects from the Cohn-Kanade DB (printed with permission from the researchers).

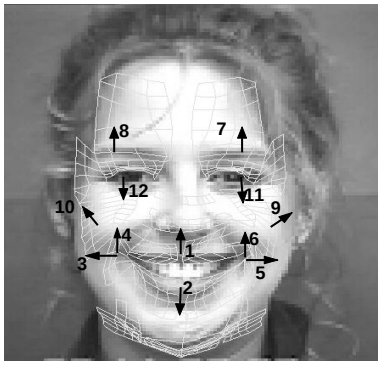

Figure 2: The facial motion measurements

The face tracking we use in our system is based on a system developed by Tao and Huang [33] called the piecewise Bézier volume deformation (PBVD) tracker.

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 (MU's). Note that they are similar but not equivalent to Ekman's AU's [14], 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 MU's used in the face tracker are shown in Figure 2. The MU's are used as the features for the Bayesian network classifiers.

5. EXPERIMENTS

We show results of Bayesian network classifiers learned with labeled and unlabeled data for facial expression recognition. We use our non-rigid face tracking system presented in Section 4 and extract features in the form of 12 facial motion units. 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 is labeled. We removed the labels of most of the training data and learned the classifiers with the different approaches discussed in Section 3.

The first database was collected by Chen and Huang [7] 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 video sampling rate was 30 Hz, and a typical emotion sequence is about 70 samples long ($\sim$2s). Figure 3 (upper row) shows one frame of each subject from this database.

The second database is the Cohn-Kanade database [22] 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. For each subject there is at most one sequence per expression with an average of 8 frames for each expression. Figure 3 (lower row) shows some examples used in the experiments.

A summary of both databases is presented in Table 1. We measure the accuracy with respect to the classification result of each frame, where each frame in the video sequence was manually labeled to one of the expressions (including Neutral). This manual labeling can introduce some 'noise' in our classification because the boundary between Neutral and the expression of a sequence is not necessarily optimal, and frames near this boundary might cause confusion between the expression and the Neutral.

The results are shown in Table 2, 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

Table 1: Summary of the databases and the description of the datasets.

Database	# of Subjects	Overall # of sequences per expression	# of sequences per subject per expression	average # of frames per expression	Train		Test
					# labeled	# unlabeled	
Chen-Huang DB	5	30	6	70	300	11982	3555
Cohn-Kanade DB	53	53	1	8	200	2980	1000

Table 2: 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	NB-L	NB-LUL	TAN-L	TAN-LUL	SSS-LUL
Chen-Huang	71.25±0.75%	58.54±0.81%	72.45±0.74%	62.87±0.79%	74.99±0.71%
Cohn-Kanade	72.50±1.40%	69.10±1.44%	72.90±1.39%	69.30±1.44%	74.80±1.36%

(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 [6] for a detailed analysis).

6. CONCLUSION

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

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

In closing, it is possible to view some of the components of this work independently of each other. The theoretical results of Section 2 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. Facial expression recognition 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 application to an image understanding problem we are interested in solving by learning with labeled and unlabeled data.

Acknowledgments

This work has been partly supported by HP-Labs. We thank Alex Bronstein and Marsha Duro for their suggestions and comments. We thank Marcelo Cirelo for his help, Jeffery Cohn for the use of the facial expression database, and Michael Lew for discussions on various parts of this work. We coded our own classifiers in Java, using the libraries of the JavaBayes system (freely available at http://www.cs.cmu.edu/˜javabayes). This work has been supported in part by the National Science Foundation Grants CDA-96-24396 and IIS-00-85980. The work of Ira Cohen has been supported by a Hewlett Packard fellowship.

7. REFERENCES

[1] S. Baluja. Probabilistic modelling for face orientation discrimination: Learning from labeled and unlabeled data. In *NIPS*, 1998.

[2] K. Bennett and A. Demiriz. Semi-supervised support vector machines. In *NIPS*, pages 368–374, 1998.

[3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100, 1998.

[4] M. Brand. An entropic estimator for structure discovery. In *NIPS*, 1998.

[5] R. Bruce. Semi-supervised learning using prior probabilities and EM. In *IJCAI Workshop on Text Learning*, 2001.

[6] V. Castelli. *The relative value of labeled and unlabeled samples in pattern recognition*. PhD thesis, Stanford, 1994.

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

[8] J. Cheng, R. Greiner, J. Kelly, D. A. Bell, and W. Liu. Learning bayesian networks from data: An information-theory based approach. In *The Artificial Intelligence Journal, Volume 137*, pages 43–90, 2002.

[9] C.K. Chow and C.N. Liu. Approximating discrete probability distribution with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467, 1968.

[10] I. Cohen. *Semi-supervised learning of classifiers with application to human computer interaction*. PhD thesis, University of Illinois at Urbana-Champaign, Dept. of Electrical Engineering, 2003.

[11] I. Cohen, N. Sebe, F. Cozman, M. Cirelo, and T Huang. Learning bayesian network classifiers for facial expression recognition using both labeled and unlabeled data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[12] I. Cohen, N. Sebe, A. Garg, L. Chen, and T.S. Huang. Facial expression recognition from video sequences: Temporal and static modelling. *Computer Vision and Image Understanding*, 91(1-2):160–187, 2003.

[13] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

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

[15] J.H. Friedman. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997.

[16] N. Friedman. The Bayesian structural EM algorithm. In *UAI*, pages 129–138, 1998.

[17] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2):131–163, 1997.

[18] R. Ghani. Combining labeled and unlabeled data for multiclass text categorization. In *ICML*, 2002.

[19] D. Goleman. *Emotional Intelligence*. Bantam Books, 1995.

[20] R. Greiner and W. Zhou. Structural extension to logistic regression: discriminative parameter learning of belief net classifiers. In *UAI02*, 2002.

[21] D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report 95-06, Microsoft Research, 1995.

[22] T. Kanade, J. Cohn, and Y. Tian. Comprehensive database for facial expression analysis, 2000.

[23] A.K. McCallum and K. Nigam. Employing EM in pool-based active learning for text classification. In *ICML*, pages 350–358, 1998.

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

[25] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39:103–134, 2000.

[26] T.J. O'Neill. Normal discrimination with unclassified obseravations. *Journal of the American Statistical Association*, 73(364):821–826, 1978.

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

[28] J. Ratsaby and S.S. Venkatesh. Learning from a mixture of labeled and unlabeled examples with parametric side information. In *COLT*, pages 412–417, 1995.

[29] P. Salovey and J.D. Mayer. Emotional intelligence. *Imagination, Cognition, and Personality*, 9(3):185–211, 1990.

[30] N. Sebe, I. Cohen, A. Garg, M.S. Lew, and T.S. Huang. Emotion recognition using a Cauchy naive Bayes classifier. In *International Conference on Pattern Recognition*, 2002.

[31] M. Seeger. Learning with labeled and unlabeled data. Technical report, Edinburgh University, 2001.

[32] B. Shahshahani and D. Landgrebe. Effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Transactions on Geoscience and Remote Sensing*, 32(5):1087–1095, 1994.

[33] H. Tao and T.S. Huang. Connected vibrations: A modal analysis approach to non-rigid motion tracking. In *CVPR*, pages 735–740, 1998.