Constrained Maximum Likelihood Learning of Bayesian Networks for Facial Action Recognition

Cassio P. de Campos[1], Yan Tong[2], and Qiang Ji[1]

[1] Electrical, Computer and Systems Eng. Dept.
Rensselaer Polytechnic Institute
Troy, NY, USA
[2] Visualization and Computer Vision Lab
GE Global Research Center
Niskayuna, NY, USA

Abstract. Probabilistic graphical models such as Bayesian Networks have been increasingly applied to many computer vision problems. Accuracy of inferences in such models depends on the quality of network parameters. Learning reliable parameters of Bayesian networks often requires a large amount of training data, which may be hard to acquire and may contain missing values. On the other hand, qualitative knowledge is available in many computer vision applications, and incorporating such knowledge can improve the accuracy of parameter learning. This paper describes a general framework based on convex optimization to incorporate constraints on parameters with training data to perform Bayesian network parameter estimation. For complete data, a global optimum solution to maximum likelihood estimation is obtained in polynomial time, while for incomplete data, a modified expectation-maximization method is proposed. This framework is applied to real image data from a facial action unit recognition problem and produces results that are similar to those of state-of-the-art methods.

1 Introduction

Graphical models such as Bayesian Networks are becoming increasingly popular in many applications. During the last few years, the adoption of Bayesian networks in areas of computer vision and pattern recognition has strongly increased. Issues of the most important journals are dedicated to this matter, for instance the *Special Issue on Probabilistic Graphical Models in Computer Vision* [1] of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and the *Special Issue on Probabilistic Models for Image Understanding* [2] of the *International Journal of Computer Vision*.

Latest research uses Bayesian networks for representing causal relationships in facial expression recognition, active vision, image segmentation, visual surveillance, pattern discovery, activity understanding, amongst others. For example, Delage et al. [3] use Bayesian networks to automatically recover 3D reconstructions from single indoor images. Zhou et al. [4] apply Bayesian networks for visual

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part III, LNCS 5304, pp. 168–181, 2008.

tracking. Mortensen et al. [5] present a semi-automatic segmentation technique based on a Bayesian network constructed from a watershed segmentation. Zhang et al. [6] use Bayesian networks for modeling temporal behaviors of facial expressions in image sequences. Tong et al. [7] present a Bayesian network to recognize facial action units.

A Bayesian network encodes a joint probability distribution for its variables in a very compact graph structure, relying on a factorization in local conditional probability distributions for efficient inferences. Parameter learning is the problem of estimating probability measures of conditional probability distributions given the structure of the network. Many parameter learning techniques depend heavily on training data. Ideally, with sufficient data, it is possible to learn parameters by standard statistical analysis like maximum likelihood estimation. In many real-world applications, however, data are either incomplete or scarce, which can cause inaccurate parameter estimation. Incompleteness means that some parameter values are missing in the data, while scarceness means that the amount of training data is small. Most of computer vision problems just mentioned have to deal with scarce and incomplete data. Methods for improving parameter learning will certainly benefit many of such applications.

When data are incomplete, Expectation-Maximization (EM) [8] algorithm is often used. Even with incomplete and scarce data, qualitative knowledge about parameters is usually available, and such knowledge might be employed to improve estimations. In this paper, we propose a framework based on non-linear convex optimization to solve the parameter learning problem by combining quantitative data and domain knowledge in the form of qualitative constraints. Many types of qualitative constraints are treated, including range and relationship constraints [9], influences and synergies [10,11], non-monotonic constraints [12], weak and strong qualitative constraints [13,14].

Experiments with facial expression recognition and real image data show the benefits that qualitative constraints can impose to parameter learning and classification accuracy. Facial expressions are very important in non-verbal human communication [15]. Based on facial action units [16], it is possible to detect and measure a large number of facial expressions by virtually observing a small set of discernible muscular movements [15]. We employed a Bayesian network where nodes are associated to action units and links describe relations among them. Parameter learning is conducted using real image data and qualitative relations from experts for both complete and incomplete datasets. Although we use a simpler model and fewer training data than state-of-the-art algorithms do, inferences with our networks display recognition rates that are comparable to other results [7,17,18,19]. This indicates that our parameter learning procedure achieves high accuracy. We emphasize that we employ the proposed methods to a facial action unit recognition problem, but they are general and can be applied to a wide range of problems, as long as qualitative knowledge is available and data are scarce (it is also possible to work with large amount of data, but in those cases a standard maximum likelihood might be enough).

Section 2 comments on some related work. Section 3 introduces our notation for Bayesian networks, describes the problem of parameter learning, and details the qualitative constraints, specified by domain experts, that can guide the learning process. Then, for scarce but complete data, we describe a simple but effective procedure to solve parameter learning by reformulating the problem as a constrained convex optimization problem, which ensures global optimality in polynomial time (Section 4). For incomplete data, we describe a constrained EM idea by adding constraints to the maximization step, and iteratively solve the learning problem. Section 5 presents some experiments with real image data from a facial action unit recognition problem. Section 6 concludes the paper and indicates paths for future work.

2 Related Work

Domain knowledge can be classified as quantitative and qualitative, which describes the explicit quantification of parameters and approximate characterizations, respectively. Both are useful for parameter learning, but quantitative knowledge has been widely used while qualitative relations among parameters have not been fully exploited in many domains. Here we focus our attention on related work using qualitative relations. Parameter learning is a well explored topic and we suggest Jordan's book [20] for a broader view.

Concerning the use of qualitative relations, Wittig et al. [21] and Altendorf et al. [22] present methods to integrate qualitative constraints by introducing penalty functions to the log likelihood criterion. Weights for the penalty functions often need be manually tuned, which strongly rely on human knowledge about such weights. Feelders and Van der Gaag [23] incorporate some simple inequality constraints in the learning process, but they assume that all the variables are binary. Niculescu et al. [9,24] derive closed form solutions for the maximum likelihood estimation supposing some predefined types of constraints. However, the constraints used in all those methods are restrictive in the number of parameters and involvement of distinct distributions (usually there is no overlap between parameters of different constraints and constraints are restricted to single distributions). There are very restricted cases where parameters and constraints can involve distinct distributions. Even simple cases such as influences of Qualitative Probabilistic Networks [10] are not addressed. de Campos and Cozman [25] formulate the learning problem as a constrained optimization problem. However, they are restricted to complete datasets and apply non-convex optimization. We describe a general learning procedure that deal with a wider range of constraints and still find the global optimum solution in polynomial time.

3 Problem Definition

A Bayesian network (or BN) represents a single joint probability density over a collection of random variables. We assume throughout that variables are categorical; variables are uppercase and their assignments are lowercase.

Definition 1. *A Bayesian network is a triple* $(G, \mathcal{X}, \mathcal{P})$*, where:* $G = (V_G, E_G)$ *is a directed acyclic graph, with* V_G *a collection of vertices associated to random variables* $\mathcal{X}$ *(a node per variable), and* E_G *a collection of arcs;* $\mathcal{P}$ *is a collection of conditional probability densities* $p(X_i|PA_i)$ *where* PA_i *denotes the parents of* X_i *in the graph (*PA_i *may be empty), respecting the relations of* E_G*.*

In a BN every variable is conditionally independent of its non-descendants given its parents (Markov condition). This structure induces a joint probability distribution by the expression $p(X_1, \ldots, X_n) = \prod_i p(X_i|PA_i)$. We focus on parameter learning in a BN where structure is known in advance. Let r_i be the number of discrete categories of X_i, q_i the number of distinct assignments to PA_i (that is, $q_i = \prod_{X_t \in PA_i} r_t$) and θ be the entire vector of parameters such as $\theta_{ijk} = p(x_i^k|pa_i^j)$, where $i = 1, \ldots, n$, $j = 1, ..., q_i$ and $k = 1, ..., r_i$. Each j in pa_i^j defines a configuration to the parents of X_i. Whenever necessary and for ease of expose, we use the notation $\theta_{ijk} = \theta_{i\{x_{i_1}^{k_1}, \ldots, x_{i_t}^{k_t}\}k}$ meaning the parameter $p(x_i^k|x_{i_1}^{k_1}, \ldots, x_{i_t}^{k_t})$. We also define an order for the states of each variable X_i such that $x_i^1 < x_i^2 < \ldots < x_i^{r_i}$ (if necessary, we exchange positions of states).

3.1 Learning Parameters of a BN

Given a dataset $D = \{D_1, \ldots, D_N\}$, with $D_t = \{x_{1,t}^{k_1}, \ldots, x_{n,t}^{k_n}\}$ a sample of all BN nodes, the goal of parameter learning is to find the most probable values for θ. These values best explain the dataset D, which can be quantified by the log likelihood function $\log(p(D|\theta))$, denoted $L_D(\theta)$. Assuming that samples are drawn independently from the underlying distribution and based on conditional independence assumptions of BNs, we have $L_D(\theta) = \log \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{n_{ijk}}$, where n_{ijk} indicates how many elements of D contain both x_i^k and pa_i^j.

If the dataset D is complete, Maximum Likelihood (ML) estimation method can be described as a constrained optimization problem, i.e. maximize $L_D(\theta)$ subject to simplex equality constraints: $\forall_{i=1,\ldots n} \forall_{j=1\ldots q_i} \quad g_{ij}(\theta) = \sum_{k=1}^{r_i} \theta_{ijk} - 1 = 0$, where $g_{ij}(\theta) = 0$ imposes that distributions defined for each variable given a parent configuration sums one over all variable states. This problem has its global optimum solution at $\theta_{ijk} = \frac{n_{ijk}}{n_{ij}}$, where $n_{ij} = \sum_{k=1,\ldots,r_i} n_{ijk}$.

3.2 Qualitative Constraints

Standard likelihood estimations are usually enough if we have enough data. However, when small amount of data is available, the likelihood function may not produce reliable estimations for the parameters.

Example 2. *Suppose a BN with three binary variables (with categories* x_i^1, x_i^2*) and the following simple graph:* $X_1 \rightarrow X_2 \leftarrow X_3$*. Suppose further that we have the dataset* $D = \{D_1, D_2\}$*, with* $D_1 = \{x_1^1, x_2^1, x_3^1\}$ *and* $D_2 = \{x_1^2, x_2^2, x_3^2\}$*. Using the ML estimation, we have the posterior probabilities* $\theta_{101} = \theta_{102} = \theta_{301} = \theta_{302} = 0.5$ *and* $\theta_{2j_1 1} = \theta_{2j_2 2} = 1$*, with* $j_1 \doteq \{x_1^1, x_3^1\}$*,* $j_2 \doteq \{x_1^2, x_3^2\}$*,* $j_3 \doteq \{x_1^2, x_3^1\}$*,* $j_4 \doteq \{x_1^1, x_3^2\}$*. Posterior probability distributions* $\theta_{2j_3 k}$ *and* $\theta_{2j_4 k}$ *can not be estimated as no data about such configurations are available.*

Situations like in Example 2 could be alleviated by inserting quantitative prior distributions for the parameters. However, acquiring such quantitative prior information may not be an easy task. An incorrect quantitative prior might lead to bad estimation results. For example, standard methods apply quantitative uniform priors. In this case, if no data are present for a given parameter, then the answer would be 0.5, which may be far from the correct value. A path to overcome this situation is through qualitative information. Qualitative knowledge is likely to be available even when quantitative knowledge is not, and tends to be more reliable. For example, someone hardly will make a mistake about the qualitative relation between sizes of the Earth and the Sun; almost everyone will fail to specify a quantitative ratio (even approximate).

Example 3. *Suppose, in addition to Example 2, that the following two constraints are known:* $\theta_{302} + \theta_{2j_31} \leq 0.7$ *and* $\theta_{2j_11} \leq \theta_{2j_42}$*. With this knowledge, it is likely that* $\theta_{2j_31} \leq 0.2$ *and* $\theta_{2j_42} = 1$*, reducing the space of possible parameterizations and alleviating the problem with scarce quantitative data.*

We define a very general constraint as basis for our methods: *linear relationship constraints* define linear relative relationships between sets of weighted parameters and numerical bounds.

Definition 4. *Let* θ_A *be a sequence of parameters,* α_A *a corresponding sequence of constant numbers and* α *also a constant. A* linear relationship constraint *is defined as*

$$h(\theta) = \sum_{\theta_{ijk} \in \theta_A} \alpha_{ijk} \cdot \theta_{ijk} - \alpha \leq 0, \tag{1}$$

that is, any linear constraint over parameters can be expressed as a *linear relationship constraint*. We describe some well-known constraints that can be specified through linear relationship constraints:

- *Qualitative influences* of Qualitative Probabilistic Networks [10]: they define some knowledge about the state of a variable given the state of another, which roughly means that observing a greater state for a parent X_a of a variable X_b makes more likely to have greater states in X_b (for any parent configuration except for X_a). Although influences over non-binary variables can be described by linear relationship constraints, we use a simple binary case to illustrate: $\theta_{bj_22} \geq \theta_{bj_12} + \delta$, where $j_k \doteq \{x_a^k, pa_b^{j_*}\}$ and j_* is an index ranging over all parent configurations except for X_a. In this case, the greater state is 2, and observing x_a^2 makes more likely to have x_b^2. Note that if these constraints hold for $\delta > 0$, the influence is said *strong* with threshold δ [14]. Otherwise, it is said *weak* for δ. A *negative influence* is obtained by replacing the inequality operator $\geq$ by $\leq$ and the sign of the δ term to negative. A *zero influence* is obtained by changing inequality to an equality.
- *Additive synergies* of Qualitative Probabilistic Networks [10]: they define a conjugate influence from two parents acting to influence the child. This means that observing the same configuration for the parents X_a and X_c of

the variable X_b makes more likely to have a greater state in X_b. An example over binary variables is: $\theta_{bj_{1,1}2} + \theta_{bj_{2,2}2} \geq \theta_{bj_{1,2}2} + \theta_{bj_{2,1}2} + \delta$, where $j_{k_a,k_c} \doteq \{x_a^{k_a}, x_c^{k_c}, pa_b^{j_*}\}$ and j_* ranges over all parent configurations not including X_a nor X_c, and $\delta \geq 0$ is a constant. This forces the sum of parameters with equal configurations for X_a and X_c to be greater than the sum of parameters with distinct configurations, for all other parent configurations. Again we have exemplified using a binary case, but synergies involving non-binary variables are also linear relationship constraints. *Negative and zero additive synergies*, as well as *strong* and *weak* versions are obtained analogously.

- *Non-monotonic* influences and synergies [26]. They happen when constraints hold only for some configurations of the parents. For example, suppose three binary variables such that X_b has X_a and X_c as parents and that $\theta_{b\{x_a^2,x_c^1\}2} \geq \theta_{b\{x_a^1,x_c^1\}2}$ holds, but $\theta_{b\{x_a^2,x_c^2\}2} \geq \theta_{b\{x_a^1,x_c^2\}2}$ can not be stated. Hence we do not have a positive influence of X_a on X_b, because it would be necessary to have both constraints valid to ensure that influence. In fact we might realize that the state of X_c is relevant for the influence. In this case, we may state a non-monotonic influence of X_a on X_b that holds when X_c is x_c^1 but not when it is x_c^2. Situational signs [13] and context-specific signs [27] are some examples of non-monotonic constraints that can be encoded as linear relationship constraints.
- *Range, intra- and inter-relationship constraints* [9]. Range constraints happen when θ_A has only one parameter θ_{ijk} and $\alpha_{ijk} = 1$. In this case the constraint becomes a upper bound constraint for θ_{ijk} (we can obtain a lower bound using negative α_{ijk} and α). If all parameters involved in a linear relationship constraint share the same node index i and parent configuration j, the constraint is called *intra-relationship constraint*. Otherwise, it is a *inter-relationship constraint*.

4 Learning through Convex Optimization

Constraints of previous section can be used to describe our knowledge. As the log likelihood function is concave (a positive sum of concave functions is also concave) and we need to maximize it, our problem is in fact a constrained convex minimization program [28]:

$$\begin{aligned} \min_{\theta} -\sum_{i,j,k} n_{ijk} \cdot \log \theta_{ijk} \qquad & \text{subject to} \\ \forall_{t=1,\ldots,m} \quad h_t(\theta) &\leq 0 \\ \forall_{i=1,\ldots n} \forall_{j=1\ldots q_i} \quad g_{ij}(\theta) &= 0 \end{aligned} \tag{2}$$

where m is the number of linear relationship constraints, and g_{ij} are the simplex constraints. To exactly solve such a convex minimization program, there are many optimization algorithms. We can use specialized interior point solvers [29] or even some general optimization ideas [30], because convex programming has the attractive property that any local optimum is also a global optimum.

Furthermore, such global optimum can be found in polynomial time in the size of input [28]. We employ the Mosek software [29] to solve our convex programs. In fact non-linear convex constraints are also allowed, as convex optimization will still find the global optimum in polynomial time. On the other hand, non-convex constraints imply in losing such properties. Hence, we allow as general constraints as possible while keeping the problem tractable.

4.1 Incomplete Data

Incomplete data means that some fields of the dataset are unknown. If the dataset is $D = \{D_1, \ldots, D_N\}$, then each $D_t \subseteq \{x_{1,t}^{k_1}, \ldots, x_{n,t}^{k_n}\}$ is a sample of some BN nodes. We say that u_t is the missing part in tuple t, that is, $u_t \cap D_t = \emptyset$ and $u_t \cup D_t$ is a complete instantiation for all BN nodes. Let U be the set of all missing data. In this case, the likelihood function $\log(p(D|\theta))$ is not a simple product anymore, and the corresponding optimization program is not convex.

A common method to overcome this situation is standard EM algorithm [8], which starts from some initial guess, and then iteratively takes two types of steps (E-steps and M-steps) to get a local maximum of the likelihood function. Particularly for discrete nodes, E-step computes the expected counts for all parameters, and M-step estimates the parameters by maximizing log likelihood function, given the counts from E-step, just like would be done with a complete dataset. EM algorithm converges to a local maximum under very few assumptions [31].

Assume θ^0 is an initial guess for the parameters, and θ^t denotes the estimation after t iterations, $t = 1, 2, \ldots$. Then, each iteration of EM can be summarized as follows:

- *E-step:* compute expectation of the log likelihood given observed data D and current estimation of parameters θ^t: $Q(\theta|\theta^t) = E_{\theta^t}[\log p(U \cup D|\theta)|\theta^t, D]$.
- *M-step:* find new parameter θ^{t+1}, which maximizes expected log likelihood computed in E-step: $\theta^{t+1} = \arg\max_{\theta} Q(\theta|\theta^t)$.

We propose to extend EM with the formulation of Program (2), that is, the M-step is performed using convex programming. So, θ^{t+1} is $\arg\max_{\theta} Q(\theta|\theta^t)$, subject to linear relationship and simplex constraints, and a polynomial time algorithm solver can be employed. Because the parameter space is convex and the enhanced M-step produces a global optimum solution for the current parameter counts, this modified EM shares convergence and optimality properties of the standard EM algorithm [31]. Although the modified EM is more time expensive than the standard EM (but still polynomial) as each M-step requires the solution of a convex optimization program (standard EM may use closed form solution for ML), we argue that, just as in standard EM where an improving solution is enough instead of an optimum one (called *Generalized EM*), we might stop the convex programming as soon as an improving solution is found.

5 Experiments

In order to test the performance of our method against standard ML estimation and standard EM algorithm given scarce and incomplete data, we use random generated networks, take one network parametrization as our "truth", and then generate samples from that network. After training the models, we apply the Kullback-Leibler (KL) divergence criterion to measure the difference between joint probability distributions induced by generated networks and distributions of true networks. We conduct experiments for datasets with 100 and 1000 samples, using random *linear relationship constraints* from 2 to 8 terms in summations. The constraints are created using the true network (so they are certainly correct) in number at most equal to the number of probability distributions in the corresponding network. For each configuration, we work with twenty random sets of data and qualitative constraints. Our results show that in most part of the cases the divergence is substantially reduced (almost 40% average reduction in the divergence) when constraints are employed, which show that they are actively used during learning. Most importantly, harder problems are most benefited: scarce incomplete data and constraints performed better than large sample sets without constraints: we could verify decrease factors greater than 100 times in the amount of data needed to achieve the same accuracy results.

We now consider the problem of recognizing facial action units from real image data [18]. Based on the Facial Action Coding System [16], facial behaviors can be decomposed into a set of action units (denoted as AUs), which are related to contractions of specific sets of facial muscles. In this work, we intend to recognize 14 commonly occurring AUs.[1] We have chosen these AUs because they appear often in the literature, so it is possible to properly compare our methods with others. There are semantic relationships among them. Some AUs happen together to show a meaningful facial expression: AU_6 (cheek raiser) tends to occur together with AU_{12} (lip corner puller) when someone is smiling. On the other hand, some AUs may be mutually exclusive: AU_{25} (lips part) never happens simultaneously with AU_{24} (lip presser) since they are activated by the same muscles but with opposite motion directions. Instead of recognizing each AU individually, a probabilistic network can be employed to explicitly model relationships among AUs [7].

A BN with 14 hidden nodes is employed, where each node is associated to an AU. States of AUs are 1 (activated) and 0 (deactivated). Figure 1 depicts the structure of the BN. Note that every link between nodes has a sign, which is provided by a domain expert. Signs indicate whether there is positive or negative qualitative influence between AUs and will be commented later. For example, it is difficult to do AU_2 (outer brow raiser) alone without performing AU_1 (inner brow raiser), but we can do AU_1 without AU_2. Hence, a positive influence from AU_2 to AU_1 is stated. Furthermore, 14 measurement nodes (unshaded in Figure 1,

[1] AU_1 (inner brow raiser), AU_2 (outer brow raiser), AU_4 (brow lowerer), AU_5 (upper lid raiser), AU_6 (cheek raiser and lid compressor), AU_7 (lid tightener), AU_9 (nose wrinkler), AU_{12} (lip corner puller), AU_{15} (lip corner depressor), AU_{17} (chin raiser), AU_{23} (lip tightener), AU_{24} (lip presser), AU_{25} (lips part), and AU_{27} (mouth stretch).

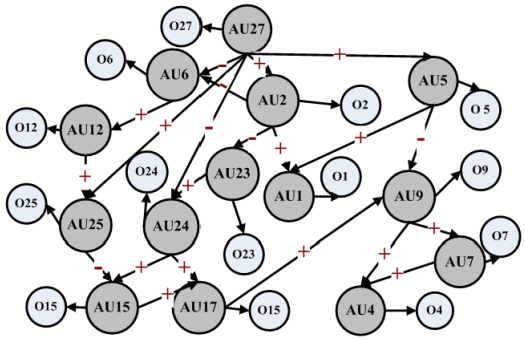

Fig. 1. Network for the AU recognition problem

one for each AU) represent classification results derived from computer vision techniques. Links between AU and measurement nodes represent uncertainties in classifications. To obtain the measurement for each AU, first the face and eyes are detected in the images, and the face region is extracted and normalized based on the detected eye positions. Then each AU is detected individually by a two-class AdaBoost classifier with Gabor wavelet features [32]. The output of the AdaBoost classifier is employed as the AU measurement in the BN model.

To parametrize the BN, training data is needed. However, it may be difficult to get enough training data to learn these parameters. The effort for training human experts and manually labeling the AUs is expensive and time consuming, and the reliability of manually coding AUs is inherently attenuated by the subjectivity of human coder. Furthermore, some AUs rarely occur. Thus, the training data can be incomplete, biased and scarce, which may cause low learning accuracy. Even though quantitative data are very important, combining them with qualitative knowledge may improve learning accuracy. Sometimes it is easier to derive qualitative relations between AUs than to fully label the data.

Parameter learning is performed using qualitative influences obtained from experts. They are described in Figure 1 (positive and negative signs mean positive and negative influences, respectively) and processed using linear relationship constraints. They are mainly based on physiological aspects:

- *Mouth stretch* increases the chance of *lips apart*; it decreases the chance of *cheek raiser and lid compressor* and *lip presser*.
- *Cheek raiser and lid compressor* increases the chance of *lip corner puller*.
- *Outer brow raiser* increases the chance of *inner brow raiser*.
- *Upper lid raiser* increases the chance of *inner brow raiser* and decreases the chance of *nose wrinkler*.
- *Nose wrinkler* increases the chance of *brow lowerer* and *lid tightener*.
- *Lip tightener* increases the chance of *lip presser*.
- *Lip presser* increases the chance of *lip corner depressor* and *chin raiser*.

We further extract some generic constraints: AU_{27} has small probability of happening, so $p(\mathrm{AU}_{27} = 1) \leq p(\mathrm{AU}_{27} = 0)$; if AU_i has more than one parent

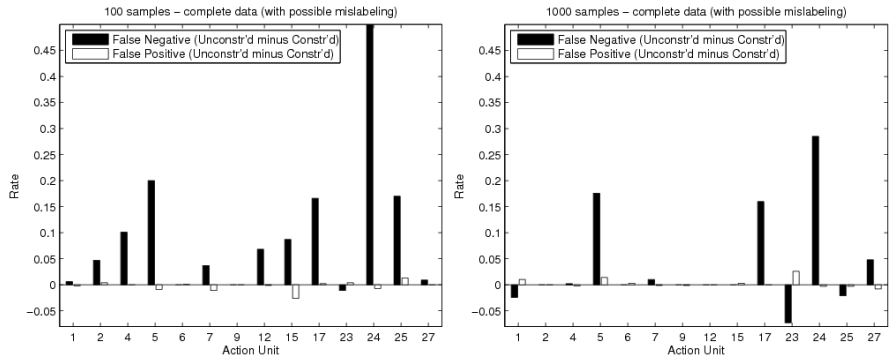

Fig. 2. Difference between unconstrained and constrained percentage rates of false negative and false positive alarms for AU recognition with complete data (but with possible mislabeling). 100 samples were used in the left graph and 1000 samples in the right graph.

node and all of them have positive influence, then $p(\mathrm{AU}_i = 1|pa(\mathrm{AU}_i) = 1) \geq 0.8$, where $pa(\mathrm{AU}_i) = 1$ means the configuration where all parents are present; if AU_i has more than one parent node and all of them have negative influence, then $p(\mathrm{AU}_i = 1|pa(\mathrm{AU}_i) = 1) \leq 0.2$. Note that these numerical assessments are conservative, as we expect the real probabilities to be greater than 0.8 (or 0.2, respectively). Conservative assessments are much more likely to be valid. Furthermore, a domain expert provide ranges (usually tight) for $p(O_i = 1|\mathrm{AU}_i = 1)$ and $p(O_i = 0|\mathrm{AU}_i = 0)$, which represent accuracy of classifiers.

The 8000 images used in experiments are collected from Cohn and Kanade's DFAT-504 database [33]. We work with three datasets: one generated from computer vision measurements (used as evidence for testing) and two from human labeling (used for training), where one is complete (but with possible incorrect labels) and other is incomplete (uncertain labels were removed). Thus, in some sense, incomplete data is more precise. We consider training data with 100 and 1000 samples. Testing is performed over 20% of the data (not chosen for training). This database was chosen because of its size and the existence of results in the literature, so as our approach can be fully compared to others and amongst different amounts of data.

Figure 2 shows the recognition results for complete data. For each AU, black bars indicate the percentage difference between false negative rates of standard and constrained ML (a positive result means that the constrained version is better). White bars are differences between false positive percentages. The accuracy using qualitative constraints is improved, specially with scarce data. For 100 samples, the average false negative using standard ML is 28%, with an average false positive of 6.5%. The constrained version obtains 17.8% of false negative, with 6.6% of false positive. We have a 10.2% improvement in the false negative rate, without considerable increase in the false positive rate. With 1000 samples, standard ML has 20.8% of false negative and 6.7% of false positive, while the

constrained version has 16.8% and 6.4%, respectively. The decrease is 4% in false negative, with also decrease in the false positive rate. Moreover, we emphasize that more than 3000 samples (without constraints) are needed to achieve the same average results as those of 100 samples and constrains (reduction greater than 30 times in the amount of data).

Figure 3 shows results for incomplete data, using standard and constrained EM. Black bars indicate differences between false negative rates while white bars are differences between false positive rates. Again, the constrained version obtains better overall results. For 100 samples, average false negative rate using standard EM is 16.7%, with a false positive of 7.1%. The constrained version obtains false negative rate of 15.3%, with 6.8% of false positive. So, we have a 1.4% improvement in the false negative rate, with also improvement in the false positive rate. With 1000 samples, standard EM has 16.6% of average false negative and 6.4% of average false positive, while the constrained version obtains 14.8% and 6.5%, respectively. This represents an overall recognition rate (percentage of correctly classified cases) of 93.7%. These last results are comparable to state-of-the-art results. For instance, Tong et at. [7] use more sophisticated models such as Dynamic Bayesian networks and employ more data for training, achieving an overall recognition rate of 93.3%. Bartlett et al. [17] reports 93.6%, and other state-of-the-art methods [32,34,35] have results with slight variance, even using more data for training. We further emphasize some points: 1) although the average rate gain is not large, we have a great gain in AU_9, because it has many missing data and constraints are fully exploited; 2) overall accuracy with incomplete data is better than that with complete data because removed labels were uncertainly labeled by human experts, so the chance of labeling error in such cases is high, and incomplete data have no such errors, which justifies the better accuracy; 3) our methods are general learning procedures that can be straightforward applied in other problems. Still, they produce results as good as those of state-of-the-art methods for the AU recognition problem.

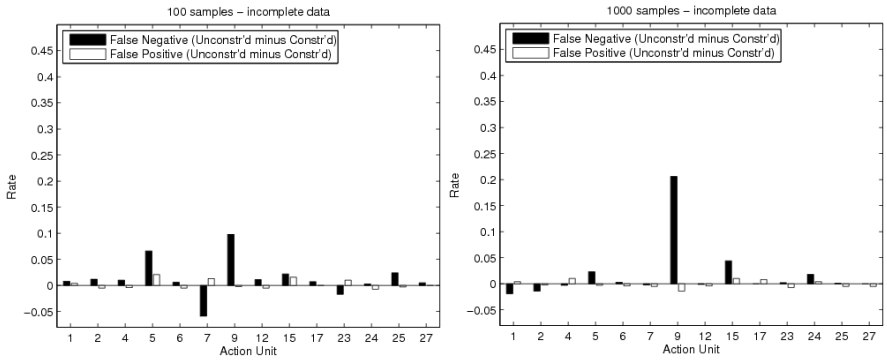

Fig. 3. Difference between unconstrained and constrained percentage rates of false negative and false positive alarms for AU recognition with incomplete data. 100 samples were used in the left graph and 1000 samples in the right graph.

We also have explored spontaneous facial expression recognition. The problem is usually much hard, as people are not posing to the camera and data are even more scarce. We have collected 1350 complete samples for training and 450 samples for testing from Belfast natural facial expression database [36] and internet repositories (e.g. Multiple Aspects of Discourse Research Lab at the University of Memphis, http://madresearchlab.org/). Using an automatically learned structure, constrained version obtains 28.4% of average false negative (decrease of 6.2% with respect to unconstrained version), with a considerably low 5.9% of average false positive (small increase of 0.6% with respect to unconstrained version). Although the relationships learned from posed facial expressions may bias the recognition for the spontaneous problem and there is a clear need to refine the system and correct some constraints by using spontaneous data, initial experiments seem promising. A deeper exploration of qualitative constraints in spontaneous datasets is left for future work.

6 Conclusion

This paper presents a framework for parameter learning when qualitative knowledge is available, which is specially important for scarce data. Even with enough data, qualitative constraints may help to guide the learning procedures. For complete data, we directly apply convex optimization to obtain a global optimum of the constrained maximum likelihood estimation, while for incomplete data, we extend the EM method by introducing a constrained maximization in the M-step. We have applied our methods to a real world computer vision problem of recognizing facial actions. For this study, constraints were elicited from domain experts. The results show that with some simple qualitative constraints from domain experts and using only a fraction of the full training data set, our method can achieve equivalent results to conventional techniques that use full training data set only. This not only demonstrates the usefulness of our work for a real world problem but also indicates its practical importance since for many applications it is often difficult to obtain enough representative training data.

Our experiments show one important application, but these techniques certainly have practical implications on other computer vision problems. Hence, future work may apply the ideas on other datasets with spontaneous facial expressions for action recognition and also on other problems such as image segmentation and body tracking. Besides that, we plan to explore other properties of the problem structure to develop and improve learning ideas based on nonlinear optimization procedures. Although the idea of using convex optimization for solving parameter learning with qualitative constraints may seem simple, we know no deep investigation of such properties has been conducted. We see the simplicity of the methods as an important characteristic, because they can be promptly applied to many real problems. While many proposals in the literature try to find specialized methods that only deal with specific constraints, we propose to use convex programming as a systematic framework for parameter learning that deals with a wide range of constraints. Finally, some words about

feasibility and the use of wrong constraints are worth mentioning: if constraints are valid, unfeasible problems never happen. We have assumed that constraints are valid, which is reasonable as we have worked with very general constraints. A systematic study of possible wrong constraints is left for future work.

References

1. Ji, Q., Luo, J., Metaxas, D., Torralba, A., Huang, T., Sudderth, E. (eds.): Special Issue on Probabilistic Graphical Models in Computer Vision, IEEE Transactions on Pattern Analysis and Machine Intelligence (2008), http://www.ecse.rpi.edu/homepages/qji/PAMI_GM.html
2. Triggs, B., Williams, C. (eds.): Special Issue on Probabilistic Models for Image Understanding. International Journal of Computer Vision (2008), http://visi.edmgr.com/
3. Delage, E., Lee, H., Ng, A.: A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In: Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition (2006)
4. Zhou, Y., Huang, T.S.: Weighted Bayesian network for visual tracking. In: Proc. of the International Conference on Pattern Recognition (2006)
5. Mortensen, E., Jia, J.: Real-time semi-automatic segmentation using a Bayesian network. In: Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition (2006)
6. Zhang, Y., Ji, Q.: Active and dynamic information fusion for facial expression understanding from image sequence. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(5), 699–714 (2005)
7. Tong, Y., Liao, W., Ji, Q.: Facial action unit recognition by exploiting their dynamic and semantic relationships. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1683–1699 (2007)
8. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B 39(1), 1–38 (1977)
9. Niculescu, R.S.: Exploiting Parameter Domain Knowledge for Learning in Bayesian Networks. PhD thesis, Carnegie Mellon (2005) CMU-CS-05-147
10. Wellman, M.P.: Fundamental concepts of qualitative probabilistic networks. Artificial Intelligence 44, 257–303 (1990)
11. Wellman, M.P., Henrion, M.: Explaining explaining away. IEEE Transactions on Pattern Analysis and Machine Intelligence 15, 287–307 (1993)
12. van der Gaag, L.C., Bodlaender, H.L., Feelders, A.: Monotonicity in Bayesian networks. In: UAI, pp. 569–576. AUAI Press (2004)
13. Bolt, J.H., van der Gaag, L.C., Renooij, S.: Introducing situational influences in QPNs. In: Nielsen, T.D., Zhang, N.L. (eds.) ECSQARU 2003. LNCS (LNAI), vol. 2711, pp. 113–124. Springer, Heidelberg (2003)
14. Renooij, S., van der Gaag, L.C.: Enhancing QPNs for trade-off resolution. In: UAI, pp. 559–566 (1999)
15. Pantic, M., Bartlett, M.: Machine analysis of facial expressions, pp. 377–416. I-Tech Education and Publishing, Vienna, Austria (2007)
16. Ekman, P., Friesen, W.V.: Facial action coding system: A technique for the measurement of facial movement. Consulting Psychologists Press, Palo Alto (1978)

17. Bartlett, M.S., Littlewort, G.C., Frank, M.G., Lainscsek, C., Fasel, I., Movellan, J.R.: Automatic Recognition of Facial Actions in Spontaneous Expressions. Journal of Multimedia 1(6), 22–35 (2006)
18. Pantic, M., Rothkrantz, L.J.M.: Automatic analysis of facial expressions: The state of the art. IEEE Trans. Pattern Analysis and Machine Intelligence 22, 1424–1445 (2000)
19. Tian, Y., Kanade, T., Cohn, J.: Facial expression analysis. Springer, Heidelberg (2004)
20. Jordan, M. (ed.): Learning Graphical Models. The MIT Press, Cambridge (1998)
21. Wittig, F., Jameson, A.: Exploiting qualitative knowledge in the learning of conditional probabilities of Bayesian networks. In: UAI, pp. 644–652 (2000)
22. Altendorf, E., Restificar, A.C., Dietterich, T.G.: Learning from sparse data by exploiting monotonicity constraints. In: UAI, pp. 18–26 (2005)
23. Feelders, A., van der Gaag, L.C.: Learning Bayesian network parameters under order constraints. International Journal of Approximate Reasoning 42(1-2), 37–53 (2006)
24. Niculescu, R.S., Mitchell, T., Rao, B.: Bayesian network learning with parameter constraints. Journal of Machine Learning Research 7(Jul), 1357–1383 (2006)
25. de Campos, C.P., Cozman, F.G.: Belief updating and learning in semi-qualitative probabilistic networks. In: UAI, pp. 153–160 (2005)
26. Renooij, S., van der Gaag, L.C.: Exploiting non-monotonic influences in qualitative belief networks. In: IPMU, Madrid, Spain, pp. 1285–1290 (2000)
27. Renooij, S., van der Gaag, L.C., Parsons, S.: Context-specific sign-propagation in qualitative probabilistic networks. Artificial Intelligence 140, 207–230 (2002)
28. Ben-Tal, A., Nemirovski, A.: Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications. MPS/SIAM Series on Optimization. SIAM (2001)
29. Andersen, E.D., Jensen, B., Sandvik, R., Worsoe, U.: The improvements in mosek version 5. Technical report, Mosek Aps (2007)
30. Murtagh, B.A., Saunders, M.A.: Minos 5.4 user's guide. Technical report, Systems Optimization Laboratory, Stanford University (1995)
31. Wu, C.F.J.: On the convergence properties of the EM algorithm. The Annals of Statistics 11(1), 95–103 (1983)
32. Bartlett, M.S., Littlewort, G., Frank, M.G., Lainscsek, C., Fasel, I., Movellan, J.R.: Recognizing facial expression: Machine learning and application to spontaneous behavior. In: Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2nd edn., pp. 568–573 (2005)
33. Kanade, T., Cohn, J.F., Tian, Y.: Comprehensive database for facial expression analysis. In: IEEE International Conference on Automatic Face and Gesture Recognition, pp. 46–53 (2000)
34. Valstar, M.F., Patras, I., Pantic, M.: Facial action unit detection using probabilistic actively learned support vector machines on tracked facial point data. In: Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, Workshop Vision for Human-Computer Interaction (2005)
35. Tian, Y., Kanade, T., Cohn, J.: Recognizing action units for facial expression analysis. IEEE Trans. Pattern Analysis and Machine Intelligence 23(2), 97–115 (2001)
36. Douglas-Cowie, E., Cowie, R., Schroeder, M.: The description of naturally occurring emotional speech. In: Int'l Congress of Phonetic Sciences (2003)