Learning a Restricted Bayesian Network for Object Detection

Henry Schneiderman[1]

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
hws@cs.cmu.edu

Abstract

Many classes of images have the characteristics of sparse structuring of statistical dependency and the presence of conditional independencies among various groups of variables. Such characteristics make it possible to construct a powerful classifier by only representing the stronger direct dependencies among the variables. In particular, a Bayesian network compactly represents such structuring. However, learning the structure of a Bayesian network is known to be NP complete. The high dimensionality of images makes structure learning especially challenging. This paper describes an algorithm that searches for the structure of a Bayesian network based classifier in this large space of possible structures. The algorithm seeks to optimize two cost functions: a localized error in the log-likelihood ratio function to restrict the structure and a global classification error to choose the final structure of the Network. The final network structure is restricted such that the search can take advantage of pre-computed estimates and evaluations. We use this method to automatically train detectors of frontal faces, eyes, and the iris of the human eye. In particular, the frontal face detector achieves state-of-the-art performance on the MIT-CMU test set for face detection.

1. Introduction

Many classes of images have sparse structuring of statistical dependency. Each variable has strong statistical dependency with a small number of other variables and negligible dependency with the remaining ones. For example, geometrically aligned images of faces, cars, and telephones exhibit this property. Figure 1 shows empirical mutual information among wavelet variables representing frontal human face images. (Mutual information measures the strength of the statistical dependence between two variables.) Each "image" represents the mutual information values between one chosen wavelet variable, indicated by an arrow, and all the other variables in the wavelet transform. The brightness at each location indicates the mutual information between the variable at this location and the chosen variable. These examples illustrate common behavior where a given variable is statistically related with a small number of other variables.

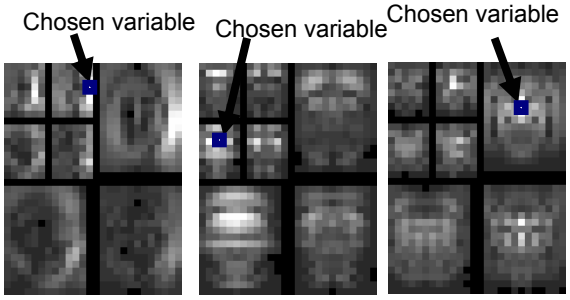
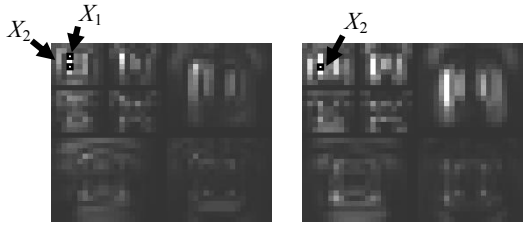


Fig. 1. Empirical mutual information among wavelet variables sampled from frontal faces images.

Sparse structuring of statistical dependency may explain the empirical success of "parts-based" methods for face detection and face recognition [1][2][3][4][5][6][7]. These methods concentrate modeling power on localized regions, in which dependencies tend to be strong.

However, statistical dependency is not always decomposable into separate "parts". Consider two variables, X_1 and X_2, that are strongly dependent on each other. Each variable may have some dependencies that are not shared by the other variable. For example, X_2 may have dependencies with a subset of variables, S_2, that only have a weak dependency with X_1 and vice versa. This situation is shown in Fig. 2 for empirical data collected from telephone images.

[1] This work was supported in part by the Advanced Research and Development Activity (ARDA)'s Video Analysis and Content Extraction Program (VACE) under contract number MDA904-03-C-1789 and the TSWG under contract N41756-03-C-4024.



Mutual Information with X_1 | Mutual Information with X_2

Fig. 2. An example of the non-decomposibility of statistical dependency over telephone images.

This type of non-decomposibility of statistical dependency can be compactly represented using the notion of conditional independence. In the case above, the probability distribution can be parameterized by conditioning on the subset, *H,* consisting of variables that share dependencies with both X_1 and X_2: $P(S_1|H)$ $P(S_2|H)P(H)$. Graphical models (Bayesian networks, Markov Random Fields, Factor Graphs, Chain graphs, mixtures of trees) provide parameterizations that can compactly capture such behavior. In particular, such models can be used for classification in a generative framework with a separate graphical model (e.g., Bayesian network) representing each class-conditional probability distribution.

In general, the problem of learning dependency structure for image classification problems has not received much attention in the computer vision community. Previous "parts-based" methods, use hand-picked "parts." In [8] we describe how we learn the simple structure of a semi-naïve Bayes classifier. In this paper, we propose a method to learn the dependency structure of a Bayesian network based classifier.

The main challenge is that learning the structure of a Bayesian network is known to be NP complete; that is, the only guaranteed optimal solution is to construct classifiers over every possible network structure and explicitly compare their performance. Moreover, the number of possible structures is super-exponential in the number of variables. Heuristic search is the only possible means of finding a solution.

Much work has focused on Bayesian scoring techniques for learning model structure of Bayesian Networks [9][10][11]. Such methods are attractive for a number of reasons and, in particular, automatically embody Occam's razor [12]. However, these methods involve computing a score for each possible model by summing probabilities of the data over all possible instantiations the model's parameters. The computational complexity of this computation is too large for image classification problems which involve a few hundred to a few thousand variables and as many as 10^6 training examples.

We propose an approach that selects a structure by seeking to optimize a sequence of two cost functions. The first optimization is over local error in the log likelihood ratio function. This function assumes every pair of variables is independent from the remaining variables. In particular, we organize the variables into a large number of candidate subsets such that this measure is minimized. This organization will restrict the final network to representing dependencies that occur only within subsets. This grouping, however, allows for efficient search in the second optimization. This optimization chooses the final network structure by minimizing a global measure of empirical classification error computed on a cross-validation set of images. Ordinarily, global measures present a complexity challenge. In particular, the cost of computing a classification error score for even one structure can be significant. It involves estimating conditional probability functions over the entire set of training images. Typically, high dimensional problems such as image classification require a large set of examples, e.g., as many as 10^6. Cross-validation cost can also be significant. Moreover, the number of possible structures in this reduced space of possible structures is quite large. We therefore restrict the final network structure to reduce these costs. In particular, we restrict the network to consist of two layers of nodes. Nodes in the top layer each consist of a single variable. Nodes in the second layer each consist of a subset of variables. By imposing this structure, it is assumed that the parents of any node in the second layer are statistically independent. Under this restriction, the overall network can be written in the following form:

$$(1)\quad P(X_1,...,X_n) = \frac{P(S_{j(1)})P(S_{j(2)})...P(S_{j(r)})}{[P(X_1)]^{\alpha_1}[P(X_2)]^{\alpha_2}...[P(X_n)]^{\alpha_n}}$$

where S_1. . . S_q are subsets of the input variables generated in the first optimization:

$$S_1, S_2, ..., S_q \subset \{X_1, ..., X_n\}$$

$$j(k) \in \{1, 2, ..., q\}$$

Each α_k corresponds to the number of occurrences of the given variable in the r subsets. For example, if X_k occurs in 3 subsets, then α_k would equal 2. In general, the denominator could be thought of as a term that corrects for "over-counting"; that is, the occurrence of a variable in more than one subset.

This form makes it possible to find the final structure using pre-computed estimates and evaluations over the candidate subsets and the individual variables.

We use this method in the context of object detection. Object detection is the task of finding instances of the given object anywhere in an image and at any size. Therefore, to perform detection, a detector exhaustively

scans a classifier over the input image and resized versions of the input image (e.g. with a scale factor of 1.189). The classifier operates on fixed size input "window", e.g., 32x24 for frontal faces, and allows for a limited amount of variation in size and alignment of the object within this window.

We use this approach to construct classifiers for detecting several types of objects: frontal faces, eyes, and irises. These detectors perform robustly with a high detection rate and low false alarm rate in unconstrained scenery over a wide range of variation in background scenery and lighting.

2. Construction of the Classifier

The classifier takes the following general form when written as a log likelihood ratio test:

$$(2) \qquad f(X_1,\ldots,X_n) = \log \frac{P(X_1,\ldots,X_n|\omega_1)}{P(X_1,\ldots,X_n|\omega_2)} > \lambda$$

$X_1. \ . \ . \ X_n$ are the input variables. For the experiments described in this paper, the input is a wavelet transform of a gray-scale image window. ω_1 and ω_2 indicate the two classes. For the problem of object detection, the classes are "object" and "non-object" where the non-object class represents all possible visual scenery that does not contain the object. For example, ω_1 may correspond to face and ω_2 may correspond to "non-face." This classifier chooses class ω_1 if $f(X_1,\ldots,X_n) > \lambda$. Otherwise, it chooses class ω_2. We construct the classifier within a fully supervised framework using geometrically aligned images of the object for class ω_1 and various non-object images for class ω_2. For more details about image pre-processing, training data, and the overall system for detection, refer to [13].

We represent each class-conditional probability distribution, $P(X_1. \ . \ . \ X_n|\ \omega_1)$ and $P(X_1. \ . \ . \ X_n|\ \omega_2)$, using a Bayesian network given by equation (1). We choose each of these distributions to have the same network structure.

There are two aspects to constructing the classifier: learning the Bayesian network structure and estimating probability distributions that form the network. In our approach, these two steps are coupled. Section 2.1 describes the selection of a pool of candidate subsets of variables. This choice will restrict the final network to modeling only intra-subset dependencies. Sections 2.2 and 2.3 describe dimensionality reduction and the estimation of probability distributions over these candidate subsets, respectively. Section 2.4 describes the selection of network in the restricted form of equation (1) using a combination of the candidate subsets.

2.1. Minimizing Local Error

This step creates a large collection of subsets of variables. This grouping enables efficient search using a global classification error metric, as we will explain in the next section.

We form these subsets by considering two types of modeling error: not modeling the dependency between two variables or not modeling a variable altogether. We evaluate the cost of these modeling errors in terms of their impact on the log-likelihood ratio function. We evaluate these errors only over pairs of input variables, (X_i, X_j) by assuming each pair, (X_i, X_j), is independent from the remaining input variables.. In particular, each error is the difference between the true log-likelihood ratio, log ($P(X_i, X_j\ |\omega_1)/P(X_i, X_j\ |\omega_2)$), and the log-likelihood ratio under the given modeling choice. We consider three possible cases with the following modeling errors:

$$(3) \quad C_1(X_i,X_j) = \sum_{x_i\in X_i, x_j\in X_j} P(x_i,x_j)\, abs[\log\frac{P(x_i,x_j\,|\,\omega_1)}{P(x_i,x_j\,|\,\omega_2)} - \log\frac{P(x_i\,|\,\omega_1)P(x_j\,|\,\omega_1)}{P(x_i\,|\,\omega_2)P(x_j\,|\,\omega_2)}]$$

$$(4) \quad C_2(X_i,X_j) = \sum_{x_i\in X_i, x_j\in X_j} P(x_i,x_j)\, abs[\log\frac{P(x_i,x_j\,|\,\omega_1)}{P(x_i,x_j\,|\,\omega_2)} - \log\frac{P(x_i\,|\,\omega_1)}{P(x_i\,|\,\omega_2)}]$$

$$(5) \quad C_3(X_i,X_j) = \sum_{x_i\in X_i} \sum_{x_j\in X_j} P(x_i,x_j)\, abs[\log\frac{P(x_i,x_j\,|\,\omega_1)}{P(x_i,x_j\,|\,\omega_2)}]$$

(Note, each of the random variables is assumed to be discrete-valued. We use upper case notation to denote the random variable and lower case notation to denote a particular instantiation of the variable; that is, each sum is over all possible values of the given random variable.) C_1 is the error in modeling the two variables, X_i and X_j, as independent. C_2 is the error of removing one variable, X_j, from the pair. C_3 is the error of removing both variables from the pair. We obtain these measures by empirically estimating the class conditional probability distributions, $P(X_i, X_j\ |\omega_1)$ and $P(X, X_j\ |\omega_2)$, for every pairings of variables, X_i, X_j.

Under these approximations, the error associated with a given choice of subsets, $G = \{S_1,. \ . \ .,S_r\}$, can be computed as:

$$(6) \quad E_l(G) = \sum_{\substack{(X_i,X_j),\forall i,j,i\neq j \\ X_i\bigcap X_j\notin S_k,\forall S_k \\ X_i\in S_k,\exists S_k \\ X_j\in S_k,\exists S_k}} C_1(X_i,X_j) + \sum_{\substack{(X_i,X_j),\forall i,j,i\neq j \\ X_j\notin S_k,\forall S_k \\ X_i\in S_k,\exists S_k}} C_2(X_i,X_j) + \sum_{\substack{(X_i,X_j),\forall i,j,i\neq j \\ X_j\notin S_k,\forall S_k \\ X_i\notin S_k,\forall S_k}} C_3(X_i,X_j)$$

where each S_k is a subset chosen from the set of input variables, $X_1. \ . \ . \ X_n$. We seek a set of candidate subsets, G, to minimize this localized error function. Our solution

first assigns the variables to n subsets using n greedy searches, where each input variable, X_i, is a seed for one search. This guarantees that every variable is initially represented in at least one subset and, therefore, there are no errors of the form C_2 or C_3. (This is a fairly reasonable way to initially optimize this function since the errors due to removing a variable tend to be greater than those of removing a dependency.) Each of these greedy searches adds new variables by choosing the one that has the largest sum of C_1 values formed by its pairing with all current members of the subset. Such a selection process will guarantee that the variables within any subset will have strong statistical dependency with each other.

A second search reduces the number of subsets to smaller collection. This search sequentially removes subsets until some desirable number, q, are remaining. At each step it removes the subset that will lead to the smallest increase in modeling error. In particular, it follows from equation (6) that the error in removing a given subset, S_k, is:

$$\sum_{\substack{X_i \in S_k, X_j \in S_k \\ X_i \cap X_j \notin S_i, \forall S_i, S_i \neq S_k}} C_1(X_i, X_j) + \sum_{\substack{X_i \in S_k, X_j \in S_k \\ X_j \notin Si, \forall S_i, S_i \neq S_k \\ X_i \in S_i, \forall S_i, S_i \neq S_k}} C_2(X_i, X_j) + \sum_{\substack{X_i \in S_k, X_j \in S_k \\ X_j \notin Si, \forall S_i, S_i \neq S_k \\ X_i \notin S_i, \forall S_i, S_i \neq S_k}} C_3(X_i, X_j)$$

In the experiments we describe later, the number of selected candidate subsets, q, ranged from 200 to 1,000. In general, computational cost is linear in the number of candidate subsets and is not prohibitive for large numbers.

Choosing the number of members of a subset is open question. Larger subsets have the potential to capture greater dependency. Subset size, however as we will see, is linked with dimensionality of the conditional probability distributions within the Bayesian network, and, therefore, size must be balanced against practical limits in representational power and limited training data. One possible way of addressing this issue is terms of VC dimension as described for a related problem in [5]. For simplicity we describe the algorithm assuming all subsets have the same number of members. However, in our experiments we consider multiple sizes (leading to an initial set of mn subsets, where m is the number of sizes) to allow for greater variety in the representation. In particular, we allow for subsets of size 4, 8, and 16.

2.2. Reducing Dimensionality over Subsets

In order to choose the final form of the Bayesian network, we will need to represent probability distributions over all candidate subsets, $P(S_1|\omega_1)$, ..., $P(S_q|\omega_1)$ and $P(S_1|\omega_2)$, ..., $P(S_q|\omega_2)$. It will not be possible represent these as full joint probability distributions since their dimensionality will be too great. Dimensionality reduction is necessary. Our provisional solution is to perform dimensionality reduction on each subset by linear projection:

$$F_k = A_k S_k \tag{7}$$

where S_k represents a random vector consisting of the subset's members and F_k is its the projection onto A_k. If we are projecting onto a l-dimensional space then A has l rows.

2.3. Estimating Probability Distributions

We estimate probability distributions, $P(F_k|\omega_1)$ and $P(F_k|\omega_2)$, for each candidate subset, S_k, where F_k is the reduced representation of S_k. In general there are no restrictions on the form of these probability distributions. For our experiments, we represent each distribution by a table. This representation discretizes each of the linear projections, F_k, to a discrete feature value by vector quantization. The probability tables are then estimated by counting the frequency of occurrence of each possible value of F_k in the training data. (See [13] for more details on the dimensionality reduction and probabilistic representation.)

We also estimate marginal probability distributions, $P(X_i|\omega_1)$ and $P(X_i|\omega_1)$ for each input variable X_1. . . X_n. These are used to represent the denominator of equation (1).

2.4. Minimizing Classification Error

This step makes the final choice of a Bayesian network in the form of equation (1) by selecting r of the q candidate subsets. This choice uses a global measure of classification error. In particular, we search for the choice that minimizes empirical classification error over a set of labeled examples. We measure performance using the area underneath the receiver operating characteristic (ROC) curve [14]. This measure accounts for the classifier's full operating range over values for the threshold, λ, in equation (2). There are two major difficulties in finding the classifier that optimizes this score. First, the combinatorial space of possible structures is enormous even after the restriction to intra-subset dependencies. In particular, each possible combination of r subsets gives rise to a different structure. Second, the cost of evaluating even one Bayesian network structure is significant involving probability estimation over the complete set of labeled examples. To minimize this cost, we restrict the network to have the form of equation (1). This form allows us to take advantage of pre-computed estimates of $P(F_k|\omega_1)$, $P(F_k|\omega_2)$, $P(X_i|\omega_1)$, and $P(X_i|\omega_1)$ to efficiently select the structure of the network.

Under the restrictions of equation (1), we use the following rules to form a candidate Bayesian network structure from a combination of candidate subsets. We treat disjoint subsets as statistically independent. For example, the network formed by two disjoint subsets, S_1 and S_2, becomes two disjoint nodes representing $P(S_1)$ and $P(S_2)$, respectively, with overall probability $P(S_1)P(S_2)$. On the other hand, we treat two intersecting subsets as conditionally independent on the common variables. For example, a Bayesian network composed from S_3 and S_4 with

$$R = S_3 \bigcap S_4 = \{X_{17}, X_{33}, X_{92}\}$$

has a structure with a third node representing $P(R)$. This node is then parent to two nodes with probabilities $P(S_3|R)$ and $P(S_4|R)$. The overall probability for this network is then:

$$(8) \qquad P(S_3, S_4) = P(S_3 \mid R)P(S_4 \mid R)P(R)$$

Under dimensionality reduction (section 2.2) this becomes:

$$(9) \quad P(S_3, S_4) \approx P(F_3 \mid R)P(F_4 \mid R)P(R) = \frac{P(F3)P(F4)}{P(R)}$$

where $F_3=A_3S_3$ and $F_4=A_4S_4$

Our method makes an independence assumption in representing the parent node, $P(R)$. This assumption gives the form of equation (1):

$$(10) \qquad P(S_3, S_4) \approx \frac{P(F_3)P(F_4)}{[P(X_{17})]^{\alpha 17}[P(X_{33})]^{\alpha 33}[P(X_{92})]^{\alpha 92}}$$

where the members of the parent, R, are represented as statistically independent.

This form greatly reduces the computational burden in searching for the final form of the Bayesian network. In particular, we pre-compute all subset probability distributions, $P(F_1|\omega_1),...,$ $P(F_q|\omega_1)$ and $P(F_1|\omega_2),...,P(F_q|\omega_2)$ and all marginal probability distributions, $P(X_1|\omega_1),...,P(X_n|\omega_1)$ and $P(X_1|\omega_2),...,P(X_n|\omega_2)$. With these estimates, we can form any Bayesian network of the form of equation (10) (or more generally, equation (1)) without any additional probability estimation. In comparison, the form of equation (9) involves much greater computation. We would have to estimate $P(R)$ in the very least; that is, we cannot obtain $P(R)$ by marginalizing $P(F_3)$ or $P(F_4)$ since they involve dimensionality reduction on the original variables. Nor would it have been possible to have originally estimated the full distributions, $P(S_3)$ and $P(S_4)$, because S_3 and S_4 are usually too high dimension.

The α's in equation (10) account for the dimensionality reduction. If there is no dimensionality reduction in F_3 and F_4 these would all equal one. We choose each α_i to represent how much of the energy of the original variable (e.g. X_{17}, etc) is represented in total across all the reduced variables, (e.g., F_3 and F_4). These α's are computed from the projection matrices, (e.g. A_3 and A_4). Each column projects of an original variable onto the reduced subspace. The energy of that variable in this subspace is the l_2 norm of this column. For each variable, we compute the sum of these energies, e_i, over the variable's projection onto all the chosen subsets. An energy greater than one means that there is some redundancy in representing the variable across these subsets. Therefore, we choose the term in the denominator to correct for this redundancy, where $\alpha_t = e_i - 1$. Otherwise, the contribution of the variable would be "over-counted."

In the general case, the probability distribution formed over many subsets is:

$$(11) \qquad P(X_1,...,X_n) = \frac{P(F_{j(1)})P(F_{j(2)})...P(F_{j(r)})}{[P(X_1)]^{\alpha 1}[P(X_2)]^{\alpha 2}...[P(X_n)]^{\alpha n}}$$
$$j(1)...j(r) \in \{1,...,q\}$$

The numerator gives the product of all r chosen subsets and the denominator could be thought of as correction for redundancy among these subsets.

Using this parameterization, the overall log-likelihood function is:

$$(12) \quad f(X) = \frac{\frac{P(F_{j(1)}|\omega_1)P(F_{j(2)}|\omega_1)...P(F_{j(r)}|\omega_1)}{[P(X_1|\omega_1)]^{\alpha 1}[P(X_2|\omega_1)]^{\alpha 2}...[P(X_n|\omega_1)]^{\alpha n}}}{\frac{P(F_{j(1)}|\omega_2)P(F_{j(2)}|\omega_2)...P(F_{j(r)}|\omega_2)}{[P(X_1|\omega_2)]^{\alpha 1}[P(X_2|\omega_2)]^{\alpha 2}...[P(X_n|\omega_2)]^{\alpha n}}}$$

We begin the search for final structure by pre-computing the probability of each labeled example with respect to all subset probability distributions, $P(F_1|\omega_1),...,$ $P(F_q|\omega_1)$ and $P(F_1|\omega_2),...,P(F_q|\omega_2)$, and all marginal distributions, $P(X_1|\omega_1),...,P(X_n|\omega_1)$ and $P(X_1|\omega_2),...,P(X_n|\omega_2)$. By pre-computing these, we do not need to explicitly evaluate any probability functions during search. We only have to combine these pre-computed scores to evaluate equation (12) for a given labeled example.

We greedily search for structure of equation (11) by incrementally combining subsets such that ROC area characteristic is maximized. This search can be performed multiple times to select multiple structures by restricting successive searches from making identical choices to previous searches. In particular, in our experiments we use a two part strategy that first finds l candidate combinations by comparing performance on training data (same images used to estimate probability distributions) then chooses the best of these by making more costly performance comparisons on cross-validation data (images that are separate from other aspects of training).

The experiments we describe in the next section use classifiers in the form of the reduced parameterization given by equation (12). However, it is usually possible to

re-derive a full Bayesian network with the given independencies and conditional independencies entailed by the choice of subsets (i.e., disjoint subsets are independent, overlapping subsets are conditionally independent in variables of overlap). We would expect such a network to be more accurate than that of equation (12) because it relaxes the impendence assumption in equation (10). (However, initial experiments using such a unrestricted Bayesian network for frontal face detection do not show much difference in performance, perhaps because another component of the system is limiting performance.) As well, it may be possible to search for further conditional independencies within each subset, using scores analogous to equations (3) - (5) but conditioned on various choices of a third variable in the subset.

3. Object Detection Experiments

We use this method of learning an approximate Bayesian network structure as part of a larger algorithm for object detection. Other aspects of this system are described in [13]. In evaluating the performance of this system, we used it to train a frontal face detector and compared it performance to other state-of-the-art algorithms on the MIT-CMU test set for face detection:

	89.7%	93.1%	94.4%	94.8%	95.7%
Bayesian Network *	1	8	19	36	56
Semi-Naïve Bayes*	6	19	29	35	46
[6]	31	65	--	--	--
[7]*	--	--	--	78	--
[16]*	--	--	65	--	--

Table 2. False alarms as a function of recognition rate on the MIT-CMU Test Set for Frontal Face Detection. * indicates exclusion of the 5 images of hand-drawn faces.

In this experiment, we also compare the performance of the approximate Bayesian network to a semi-naïve Bayes classifer [15]. In particular, equation (12) becomes a semi-naïve Bayes classifier if all the α's are set to zero. We trained this semi-naïve Bayes classifier using the method described in this paper, but under the constraints that all α are equal to zero. The main improvement in performance of the Bayesian network over the semi-naïve Bayes classifier is evident at lower recognition rates.

We used this method to train detectors for eyes, and the iris of the human eye. In Figures 3 and 4 we show some examples of eye and iris detection. Below we show false alarm rates as a function of recognition rate for different values of the detection threshold.

	84.3%	92.2%	96.1%
Bayesian Network	5	10	13
Semi-Naïve Bayes	4	16	54

Table 2. Eye Detection Results. False alarms as a function of recognition rate on a test set of 44 images.

	82.6%	95.7%	97.8%
Bayesian Network	15	60	87
Semi-Naïve Bayes	12	102	165

Table 3. Iris Detection Results. False alarms as a function of recognition rate on a test set of 26 images.

The results show an opposite trend to those for frontal face detection. In these experiments the Bayesian network outperforms semi-naïve Bayes at high detection rates and gives equivalent performance on lower detection rates.

One explanation for the superior performance of a Bayesian network over semi-naïve Bayes is that the network accounts for "over-counting" that is possible with a semi-naïve Bayes classifier.

4. Conclusion

The sparse nature of statistical dependency within image classes is well modeled by graphical probability models. In the computer vision community we are starting to see a rising interest in such models, most notably, [17][18]. To apply such models to classification problems is challenging, though, because of high-dimensionality. In the paper, we have presented an approach for learning such a classifier under a few reasonable modeling assumptions and have trained accurate detectors for the tasks of face detection, eye detection, and detection of the iris of the human eye.

References

[1] Moghaddam, B.; Pentland, A. *"*Probabilistic visual learning for object representation*." IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:7, 696 -710 (1997)

[2] Burl, M. C., Weber, M. and Perona, P. (1998). A Probabilistic Approach to Object Recognition using Local Photometry and Global Geometry. In Proc. of the *5th European Conf. On Computer Vision* , 1998.

[3] Schneiderman, H. and Kanade, T. (1998). "Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition." *IEEE Conference on Computer Vision And Pattern Recognition,* 1998.

[4] Rowley, H.A., Baluja, S. and Kanade, T. (1998) Neural Network-Based Face Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23-38.

[5] Heisele, B., T. Serre, M. Pontil and T. Poggio. "Component Based Face Detection." CVPR, 2001. (2001)

[6] Viola P. and Jones M. Rapid Object Detection Using a Boosted Cascade of Simple Features. CVPR, 2001. (2001)

[7] Roth, D., Yang, M-H., Ahuja, N. A SNoW-Based Face Detector. *NPPS-12*. (1999)

[8] Schneiderman, H. "Learning Statistical Structure for Object Detection." *Computer Analysis of Images and Patterns (CAIP), 2003*, Springer-Verlag, August, 2003.

[9] Cooper, G. F., Herskovits, E. (1992). "A Bayesian Method for the Induction of Probabilistic Networks from Data." *Machine Learning*, 9:309-347.

[10] Heckerman, D., Geiger, D., Chickering, D. H. (1995) "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data." *Machine Learning* 20(3): 197-243

[11] Friedman, N. and Koller, D. (2002) "Being Bayesian about Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks." *Machine Learning Journal*.

[12] MacKay, D. J. C. *Information Theory, Inference, and Learning Algorithms.*

[13] Schneiderman, H. "Feature-Centric Evaluation for Efficient Cascaded Object Detection." CVPR, 2004.

[14] Duda, R. O., Hart, P. E., Stork, D. G. (2001) *Pattern Classification*. John Wiley & Sons.

[15] Kononenko, I. "Semi-Naïve Bayesian Classifier." Sixth European Working Session on Learning. pp. 206-219. 1991

[16] Schneiderman, H., Kanade, T. "A Statistical Method for 3D Object Detection Applied to Faces and Cars". CVPR, 2000

[17] Murphy, K., Torralba, A. and Freeman, W. T. *"*Using the forest to see the trees: a graphical model relating features, objects, and scenes." NIPS-16, 2004

[18] Sudderth, E. B., Ihler, A. T., Freeman, W. T. and Willsky, A. S. *"*Nonparametric Belief Propagation and Facial Appearance Estimation." CVPR, 2003

Fig. 3. Eye Detection

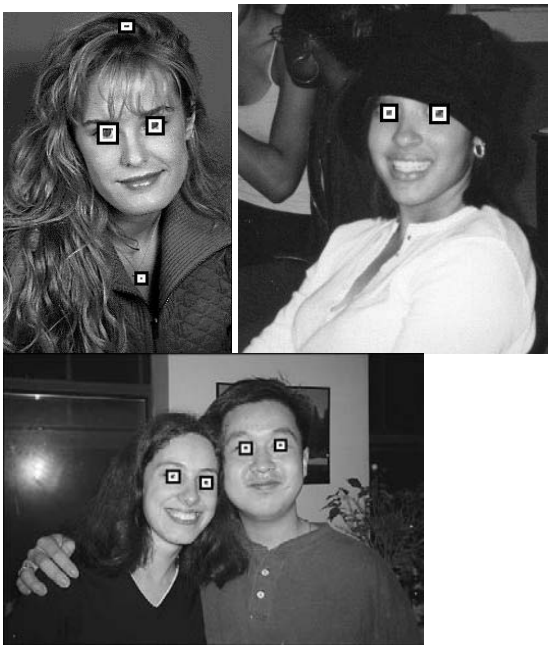


Fig. 4. Detection of the Iris of the Human Eye