Infomax Boosting

Siwei Lyu
Department of Computer Science
Dartmouth College

Abstract

In this paper, we described an efficient feature pursuit scheme for boosting. The proposed method is based on the infomax principle, which seeks optimal feature that achieves maximal mutual information with class labels. Direct feature pursuit with infomax is computationally prohibitive, so an efficient gradient ascent algorithm is further proposed, based on the quadratic mutual information, non-parametric density estimation and fast Gauss transform. The feature pursuit process is integrated into a boosting framework as infomax boosting. The performance of a face detector based on infomax boosting is reported.

1. Introduction

Many problems in computer vision rely on robust and efficient classifiers. In practice, classifiers are built from a set of labeled training data: $\{x_1^1, \cdots, x_{N_1}^1, \cdots, x_1^C, \cdots, x_{N_C}^C\}$, with x_i^c denoting the i-th training datum of class c. Boosting [13], with AdaBoost being the most prominent example, has been proved an effective way to construct high performance classifiers. The basic idea of boosting is to combine simple base classifiers (any classifier with performance better than random guesses) to form an ensemble. The decision function is constructed as an optimal linear combination of the base classifiers. In one round of training, a new base classifier is trained on weighted data, emphasizing on errors made in previous training rounds. With the new base classifier being included into the ensemble, the weights on training data are updated correspondingly and a new training round starts. In practice, boosting classifiers have shown remarkable performance and good generalization ability. A widely accepted explanation is that boosted classifiers achieve "large" classification margins [16].

As the base classifiers are the building blocks in boosting, they should be carefully designed for optimal performance. On high-dimensional input data, $x \in \mathbb{R}^d$, the base classifiers are usually constructed on features, which are low-dimensional representations of input data. Formally, a feature is the output of a feature map $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ evaluated on input data x, with $d' \ll d$. In this paper, we will focus on linear projection features in the data space, as $\phi(x) = \phi^T x$, with $\phi \in \mathcal{R}^d$ and $\phi^T\phi = 1$. Besides the computational benefits, using features may also help to incorporate domain knowledge that is hard to learn with a finite training set. Base classifiers built on a small set of highly informative features can reduce the overall complexity and increase the robustness of the boosting classifier. Intuitively, one should use the set of features that contain most of the classification information. This intuition is formally stated as the infomax principle [10], which recommends the use of features that maximize their mutual information with the class labels (the infomax features). However, direct feature pursuit by optimizing mutual information results in computationally prohibitive algorithms. The main contribution of this work is an efficient infomax feature pursuit for boosting. First, the class-dependent densities of each data class are estimated non-parametrically from the training data with kernel density estimations. These estimated densities give rise to an integration-free evaluation of quadratic mutual information, which is a measure of classification information. The gradient of the quadratic mutual information in terms of the feature mapped data can be computed analytically. Infomax feature pursuit then proceeds as a gradient ascent. The computation required in each step updating the gradient is further reduced from quadratic to linear in the number of training data with the fast Gauss transform approximation. The infomax feature pursuit is incorporated into a boosting framework as infomax boosting. Experimental results of applying infomax boosting to appearance-based face detection are given to show its efficacy in practice.

2. Infomax feature pursuit

A common measure of classification information in a projection feature is the mutual information [3] between the mapped data and the class labels:

$$\mathcal{I}(\phi^T x; c) = \sum_{c=1}^{C} \int_x p(\phi^T x, c) \, \log \, \frac{p(\phi^T x, c)}{p(\phi^T x) p(c)} \, d\phi^T x, \qquad (1)$$

which quantifies how much information is gained about c with the knowledge of $\phi^T x$. The most informative feature is naturally defined as the one having the maximal mutual information with class labels:

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 **IEEE**

$$\phi^* = \arg\max_{\phi} \mathcal{I}(\phi^T x; c). \quad (2)$$

The infomax principle suggests using such infomax features to build efficient and parsimonious classifiers.

Though theoretically appealing, finding infomax features with Equation (2) directly is computationally prohibitive, especially for high-dimensional data. This is due to two reasons:

- The functional form of the probability densities in computing mutual information are usually not known precisely.
- The computation of mutual information, Equation (1), requires integration.

In this paper these problems are tackled by using non-parametric density estimation and an alternative definition of the mutual information.

2.1. Kernel density estimation

Probability densities can be estimated non-parametrically from training data with the kernel density estimation (KDE) [4] as a linear combination of kernel functions. A commonly used kernel function is the isotropic multivariate Gaussian:

$$G_\sigma(x-\mu) = \frac{1}{(2\pi)^{d/2}\sigma^d} \exp\left(-\frac{(x-\mu)^T(x-\mu)}{2\sigma^2}\right), \quad (3)$$

with mean μ and variance σ^2. As shown in [14], with sufficient number of training data, a KDE with smooth kernels (such as Gaussian) can approximate any density to any degree of precision. In classification, the class-dependent density for the c-th class can be estimated with KDE as:

$$p(x|c) = \sum_{i=1}^{N_c} w_i^c G_\sigma(x - x_i^c), \quad (4)$$

where w_i^c is a non-negative weight associated with each training training datum of class c, satisfying that $\sum_{i=1}^{N_c} w_i^c = 1$. In the case of searching infomax linear projections, it is the class-dependent densities of $\phi^T x$ that are of interest to us. It is not hard to prove that the KDE of class-dependent density for $\phi^T x$ is simply the 1-D marginalization of the pdf given in Equation (4) as:

$$p(\phi^T x|c) = \sum_{i=1}^{N_c} w_i^c G_\sigma(\phi^T x - \phi^T x_i^c). \quad (5)$$

2.2. Quadratic mutual information

Kernel density estimation obviates explicit functional form of the class-dependent densities, yet the integration in the definition of mutual information still exists as a barrier to efficient computation. Using numerical integration results in inefficient and inelegant algorithms. In face of that, an alternative definition of mutual information is used whose combination with the kernel density estimation results in an evaluation without integration.

Note that the mutual information between the mapped data $\phi^T x$ and class label c can be equivalently viewed as the Kullback-Leibler (KL) divergence between their joint distribution and the product of marginals as:

$$\mathcal{I}(\phi^T x; c) = D(p(\phi^T x, c) \| p(\phi^T x)p(c)). \quad (6)$$

The KL divergence between a pair of densities p and q is

$$D(p\|q) = \int_x p(x) \log \frac{p(x)}{q(x)} dx. \quad (7)$$

As argued in [9], the KL divergence is not the only divergence measure of densities. By relaxing some defining conditions, other divergence measures can be used to satisfy requirements such as computational efficiency. One such measure is the quadratic divergence [9] defined as:

$$Q(p\|q) = \int_x \left(p(x) - q(x)\right)^2 dx. \quad (8)$$

$Q(p\|q)$ is nonnegative and is zero if and only if $p(x) = q(x)$ (a.e.). The use of quadratic divergence as a plausible surrogate of the KL divergence is justified by the following inequality:

$$D(p\|q) \quad \geq \quad \tfrac{1}{2} Q(p\|q)^2, \quad (9)$$

for which a proof is given in [12].

With the quadratic divergence, the quadratic mutual information [17] between the projected data and the class labels can be defined in a similar fashion as:

$$\mathcal{I}_Q(\phi^T x; c) = \sum_{c=1}^{C} \int_{\phi^T x} \big(p(\phi^T x, c) - p(\phi^T x)p(c)\big)^2 d\phi^T x. \quad (10)$$

Denote the class prior of the c-th class as P_c and $y_i^c = \phi^T x_i^c$, and plug in the KDEs of the class-dependent densities, the quadratic mutual information is computed as:

$$\begin{aligned}\mathcal{I}_Q(\phi^T x; c) =& \sum_{c=1}^{C} P_c^2 \sum_{i=1}^{N_c}\sum_{j=1}^{N_c} w_i^c w_j^c G_{\sqrt{2}\sigma}(y_i^c - y_j^c) \\ &+ \sum_{c_1=1}^{C}\sum_{c_2=1}^{C}\sum_{i=1}^{N_c}\sum_{j=1}^{N_{c'}} u_i^c v_j^{c'} G_{\sqrt{2}\sigma}(y_i^c - y_j^{c'}), \quad (11)\end{aligned}$$

where w_i^c is the associated weight of x_i^c in KDE, and

$$\begin{aligned} u_i^c &= P_c\left(\textstyle\sum_{c'=1}^{C} P_{c'}^2 - 2P_c\right) w_i^c \\ v_i^c &= P_c w_i^c \end{aligned} \quad (12)$$

A detailed derivation of Equation (11) and (12) is given in [12]. Note that the integration in Equation (10) vanishes as the result of using KDE with Gaussian kernels (see [12] for more information).

2.3. Optimization

Using the chain rule of differentiation and ignoring the constant factors, the gradient of $\mathcal{I}_Q$ with regards to ϕ is computed from Equation (11) as:

$$\frac{\partial \mathcal{I}_Q}{\partial \phi} = \sum_{c=1}^{C}\sum_{i=1}^{N_c} \frac{\partial \mathcal{I}_Q}{\partial y_i^c}\frac{\partial y_i^c}{\partial \phi} = \sum_{c=1}^{C}\sum_{i=1}^{N_c} \frac{\partial \mathcal{I}_Q}{\partial y_i^c} x_i^c, \qquad (13)$$

with the individual derivatives computed as:

$$\frac{\partial \mathcal{I}_Q}{\partial y_i^c} = P_c^2 \sum_{j=1}^{N_c} w_i^c w_j^c H_1\left(\frac{y_i^c - y_j^c}{2\sigma}\right) + \sum_{c'=1}^{C}\sum_{j=1}^{N_{c'}} u_i^c v_j^{c'} H_1\left(\frac{y_i^c - y_j^{c'}}{2\sigma}\right), \qquad (14)$$

where $H_1(x) = -2x\exp(-x^2)$ is the first degree Hermite function defined as $H_n(x) = (-1)^n \frac{d^n}{dx^n}\exp(-x^2)$. The optimal ϕ maximizing the quadratic mutual information can then be searched with a gradient ascent procedure: starting from an initial estimate ϕ_0, iteratively we compute $\phi_{k+1} = \phi_k + \eta \frac{\partial \mathcal{I}_Q}{\partial \phi}\Big|_{\phi=\phi_k}$ until convergence, with the parameter $\eta > 0$ controlling the step size. Convergence to a local maximum is guaranteed by the fact that the quadratic divergence is bounded [9].

2.4. Fast Gauss transform acceleration

However, in the straightforward gradient ascent presented in previous section, with $N = \sum_{c=1}^{C} N_c$ training data, each update step runs $\mathcal{O}(N^2)$ in time. When N is large, such computation is prohibitively expensive. The core computation in Equation (14) are examples of the "N-body" problem, where a potential function composed of a sum of pairwise interaction terms (in this case, the first degree Hermite functions) from N points is evaluated at an equally large number of locations.

The "N-body" problems can be approximated with a computation linear in the the size of training data in both time and space to a pre-specified precision. This is achieved with a variant of the fast Gauss transform (FGT) [7], which is a special case of the wider class of fast multipole methods. The basic idea is to group evaluation of interaction terms of farther apart points together and approximate them with a finite degree polynomial from the Taylor expansion of the Hermite function. Due to the limit of space, the details of the algorithm is given in [12]. The overall running complexity of the algorithm is linear in the total number of training data N[1]. The precision of the FGT approximation is controlled by the cut-off degree p, with the error between FGT approximation and the actual value upper-bounded by $\mathcal{O}\left(\sqrt{\frac{1}{p!}(\frac{1}{2})^{(p+1)}}\right)$. In our case, setting $p = 10$ results in a relative error of less than 10^{-6}.

[1]The complexity also depends on the dimension of data, the degree of approximating polynomial p, the size of the neighborhood r and the width of the kernel σ - all are constants for a given problem.

Algorithm 1 Infomax boosting

Given: training data $\{x_1^-, \cdots, x_{N_+}^-, x_1^-, \cdots, x_{N_-}^-\}$, class priors P_+ and P_- and width parameter of Gaussian kernels σ^2;
Initiate: $w_{i,1}^+ = \frac{1}{N_+}$, for $i = 1, \cdots, N_+$, and $w_{i,1}^- = \frac{1}{N_-}$, for $i = 1, \cdots, N_-$;
for $t = 1$ to T **do**
  Choose the infomax feature ϕ_t with $w_{i,t}^+$, $w_{i,t}^-$ and σ^2, Section 2.3-2.4;
  Construct KDE of class-dependent densities with Equation (16);
  Build base classifier $f_t(x, c)$, Equation (15);
  Update weights with Equation (17);
end for
Output decision function $\tanh\left(\sum_{t=1}^{T} f_t(x)\right)$

3. Infomax boosting

Infomax feature pursuit is incorporated into the boosting framework as infomax boosting. We chose to base our implementation on a variant of AdaBoost, "real AdaBoost" [5]. Following the conventions, the class labels are from $\{-1, +1\}$ and denote P_+ and P_- as the priors of the two classes.

The training of an infomax boosting classifier is an iterative process of incorporating base classifiers. At the t-th iteration, an infomax feature ϕ_t is obtained as described in the previous section. The t-th base classifier is trained on the weighted training data, projected on ϕ_t as:

$$f_t(x) = \log \frac{p_+^{(t)}(\phi_t^T x)P_+}{p_-^{(t)}(\phi_t^T x)P_-} \qquad (15)$$

with

$$p_+^{(t)}(\phi_t^T x) = \sum_{i=1}^{N_+} w_{i,t}^+ G_\sigma(\phi_t^T x - \phi_t^T x_i^+),$$
$$p_-^{(t)}(\phi_t^T x) = \sum_{i=1}^{N_-} w_{i,t}^- G_\sigma(\phi_t^T x - \phi_t^T x_i^-) \qquad (16)$$

being the KDEs of the two class dependent densities. The weight on each training datum is then adjusted according to the training error in the current step as:

$$w_{i,t+1}^+ = \frac{1}{Z_{t+1}^+} w_{i,t}^+ \exp(-f_t(x_i^+)),$$
$$w_{i,t+1}^- = \frac{1}{Z_{t+1}^-} w_{i,t}^- \exp(f_t(x_i^-)), \qquad (17)$$

where Z_{t+1}^+ and Z_{t+1}^- are normalizing constants ensuring that $\sum_{i=1}^{N_+} w_{i,t+1}^+ = 1$ and $\sum_{i=1}^{N_-} w_{i,t+1}^- = 1$. At the completion of training, the algorithm outputs a soft-threshold

decision function $f(x) = \tanh\left(\sum_{t=1}^{T} f_t(x)\right)$, from which a hard-threshold decision function is constructed as $F(x) = sgn(f(x) - \tau)$. Threshold τ is pre-chosen and is used to control the desired false positive/negative rates of the hard-threshold classifier. The overall training process is summarized in Algorithm 1.

4. Face detection

Infomax boosting is applied to the appearance-based face detection [20], whose goal is to determine if there is any face in an image and if so, locate the positions and extents of the faces. The face detector is constructed as a cascade of infomax boosting classifiers, each of which individually determines if a sub-image contains a face pattern or not.

4.1. Experimental setup

The training set of the face detector were constructed from a set of face and non-face patterns. The face patterns were collected from $2,000$ frontal facial images in the FERET database [15]. By incorporating slightly rotated, scaled and mirrored variants, the image set was further enlarged to $20,000$. Each face image was then scaled and aligned to generate patches with a base resolution of 20×20 pixels, which was further normalized to $[0, 255]$ to compensate variance in illumination. The non-face patterns were generated from $11,320$ images downloaded from `www.freefoto.com`. These images contain no human faces and spread a wide range of natural and man-made scenes. Subsequently, $10,000,000$ non-face patterns of size 20×20 pixels were generated from these images.

Similar to [11], infomax features were formed from linear combinations of an over-complete wavelet bank, which consisted of 72 oriented and scaled Gaussian and Gaussian derivative filters of first and second order (Figure 1). Specifically, each training pattern was convolved with each of the 72 filters, yielding a total 72 vectors of 400 dimension, which were treated as $28,800$ individual local features. More global image features were captured by another set of 72 similar filters with a larger width. The inner 10×10 region of each image patch was convolved with this set of filters, providing another set of $7,200$ features, yielding a total $36,000$ features for each image. The infomax features were built as the linear combinations of the top $1,000$ elementary features in the $36,000$ feature repertoire, ordered by their quadratic mutual information with the class labels, Equation (10). To avoid over-emphasizing any position in the image patch, at most two features of the same spatial location were included as elementary features. Infomax features are searched using the gradient ascent procedure (Section 2.3) with the fast Gauss transform accelerations (Section 2.4)[2]. The initial values for ϕ were the outputs of a linear discriminant analysis [4] on all training data. In finding the infomax feature, we set the cut of degree of the Hermite expansion to 10 and neighborhood size to 7. The width of the Gaussian is chosen empirically for best performance. Individual classifier is then trained with Algorithm 1.

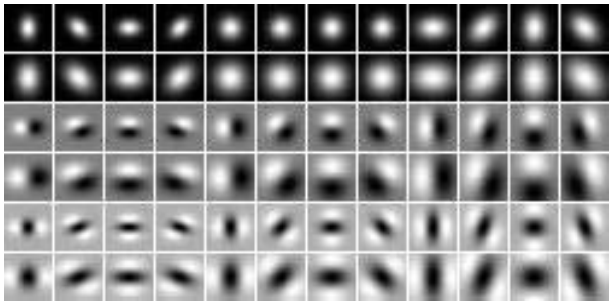

Figure 1: Filter bank used in our experiment.

Following the method proposed in [19], our face detector was constructed as a cascade of boosting classifiers. Each classifier in the cascade has a fixed false negative rate (misclassifying a face pattern as a non-face pattern), while the false positive rate (misclassifying a non-face pattern as a face pattern) and the number of base classifiers are controlled to be under pre-given thresholds. A 20×20 sub-window is classified to contain a face pattern only if it passes all classifiers in the cascade. Besides the infomax boosting method, a face detector based on a cascade of AdaBoost classifiers was also trained with the same training data. The base classifiers in the AdaBoost detector were constructed on randomly chosen elementary features. Its performance was used to compare with that of the infomax boosting detector.

The training of the classifier cascade was done in a bootstrapping way. All training data were first split into a training set and a validation set. The initial classifier was trained on the whole training set. Features were added into the classifier until either of the following conditions became true: (i) the total number of features exceeded 20; (ii) the false negative rate exceeded 0.01%; (iii) the false positive rate was over 30%. The training set of the next classifier consisted of all face patterns and the false-positives in the current training set. The overall classifier cascade had 30 classifiers, with 423 features in total.

The infomax boosting face detector was tested on the MIT+CMU face detection database, which contains 130 images and 507 labeled faces. To find all possible face patterns in a test image, the detector scanned over all 20×20 sub-windows in the images. Face patterns may also occur at different scales, so the detector also scanned a series of up-scaled and down-scaled copies of the test images. The

[2]The fast Gauss transform on Hermite kernels is implemented based on the FGT implementation downloaded from `http://www.cs.cornell.edu/aflorenc/research/fgt`.

scaling was achieved by interpolating the original image with bi-cubic splines followed by up-sampling or down-sampling. We used a set of scaling factors 1.1 apart as it led to the best performance in our experiments. These operations resulted in a total $84,638,740$ sub-windows being scanned from the MIT+CMU test set.

To merge multiple detections of the same face in different positions and scales, we employed a simple clustering of the detected face patterns similar to [19]. Specifically, the centers of all detected face patterns in an image of different spatial locations and scales were subject to k-means clusterings with varying k values. The number of distinct face patterns in the image was determined by the k value that resulted in a clustering where the maximal within cluster distance was less than 3 pixels and the minimal between cluster distance was larger than 12 pixels. Each detected face was then attribute to the closest cluster center. The locations of the final detected faces were the averages of all sub-windows in each individual cluster.

4.2. Results

Shown in Figure 2 is the plot of the testing false positive rate versus the number of features included in the face detector for both infomax boosting and AdaBoost. The infomax boosting detector tended to outperformed the AdaBoost detector: it takes 50 features to achieve a lower than 1% false positive rate in the infomax detector, while to have a similar false positive rate, the AdaBoost detector needs around 300 features. Shown in Figure 3 are the ROC curves on the testing set of both the infomax boosting detector and the AdaBoost detector. With a 10^{-6} false positive rate, it had a detection rate of 96.3%, while the AdaBoost detector had a detection rate of 85.1%. This result is also slightly better than that of [11], but such comparison is not conclusive, as the numbers do not reflect significant difference in training sets, parameters and other important aspects of the detectors. Finally, shown in Figure 4 are outputs of the infomax face detector on some of the test images in the MIT+CMU test set, where the white and black boxes are the maximal and minimal bounding boxes of of the same face detected.

We also compared the gradient update in searching the infomax features with and without fast Gauss transform acceleration. With a 10^{-9} relative error, the FGT achieved a speed up of more than 200 fold for each updating step, drastically improving the efficiency of the training process.

5. Related work

Selection of optimal features in a binary boosting classifier was first studied in [19] in the context of face detection. Features with best classification performance were chosen from a repertoire of an over-complete Haar wavelet

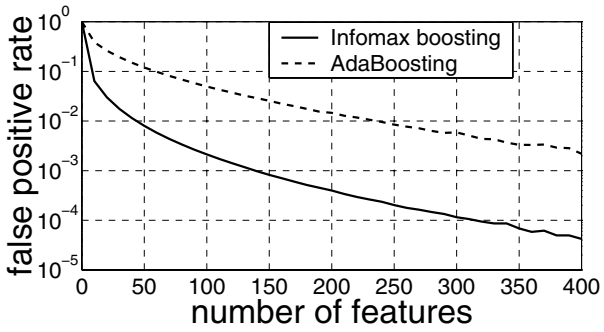

Figure 2: Curves of false alarm rate vs. the number of features included in the boosting detectors.

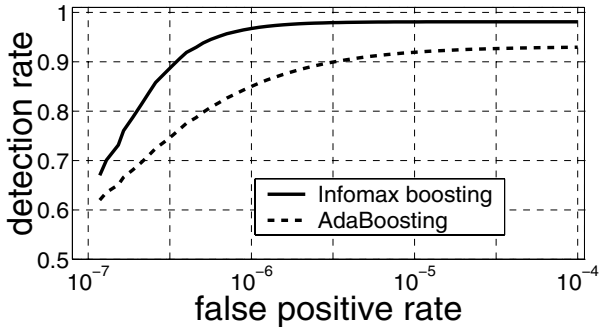

Figure 3: ROC curves of boosting face detectors on the MIT+CMU test set.

bank. Instead of choosing features from a given set optimal projection feature pursuit was combined with boosting in the framework of Kullback-Leibler (KL) boosting [11]. The discriminant ability of a feature is measured by the Kullback-Leibler divergence between the histograms of projected data, with the optimal projection found with a grid search. The KL divergence is a special case of mutual information for binary classification with equal class priors, while the mutual information criterion has no constraints on the number of classes. Moreover, the gradient ascent optimization as presented in this work is more flexible than a grid search.

The infomax principle was first proposed as a model of neural organization by Linsker [10] and has hence received much attention in pattern recognition and machine learning. Its application includes independent component analysis [1], blind source separation [2], nonlinear discriminant analysis [17], feature selection [18] and neural network [8]. The use of quadratic mutual information and KDE to compute mutual information was first proposed in [6] and further developed in [17].

6. Discussion and future work

We have discussed a new feature pursuit scheme for boosting driven by the infomax principle. One advantage of infomax feature pursuit is that there is no built-in constraint

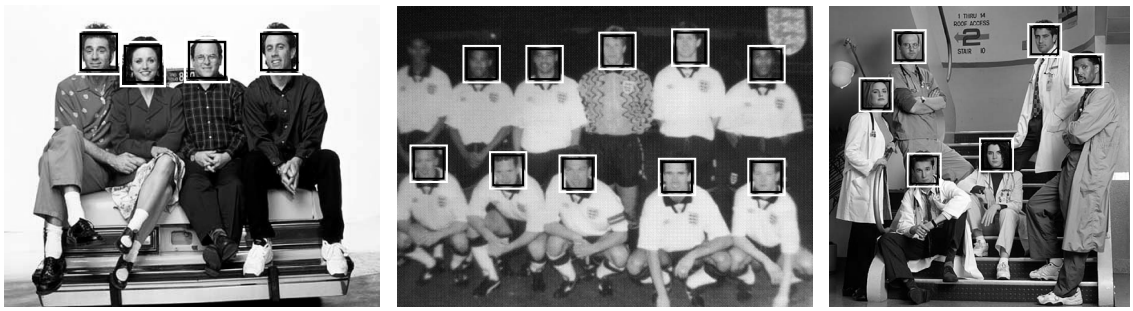

Figure 4: Outputs of infomax face detector on some images in the MIT+CMU test set.

on the number of classes in the mutual information criterion, suggesting its use in a multi-class classification. One of our future works then is to design a multi-class boosting framework to fully take this advantage.

Acknowledgment

This work was supported by Hany Farid under an Alfred P. Sloan Fellowship, an NSF CAREER Award (IIS99-83806), an NSF Infrastructure Grant (EIA-98-02068), and under Award No. 2000-DT-CX-K001 from the Office for Domestic Preparedness, U.S. Department of Homeland Security (points of view in this document are those of the authors and do not necessarily represent the official position of the U.S. Department of Homeland Security).

References

[1] A. Bell and T. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, (7):1129–1159, 1995.

[2] J-F. Cardoso. Infomax and maximum likelihood for blind separation. *IEEE Signal Processing Letters*, (4):112–114, 1997.

[3] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 1991.

[4] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, NY, 1973.

[5] J. Friedman, T. Hastie, and R. Tibishirani. Additive logistic regression: a statistical view of boosting. *Annual of Statistics*, 28(2):337–374, 2000.

[6] E. Gokcay and J. Principe. A new clustering evaluation function using Renyi's information potential. In *IEEE Intl. Conf. on Acoustic, Speech and Signal Proc. (ICASSP)*, 2000.

[7] L. Greengard and X. Sun. A new version of the fast Gauss transform. *Documenta Mathematica*, Extra Volume, ICM(III):575–584, 1998.

[8] N. Brunel J-P. Nadal and N. Parga. Nonlinear feedforward networks with stochastic outputs: infomax implies redundancy reduction. *Network: Comput. Neural Syst.*, (9):207–217, 1998.

[9] J. Kapur. *Measures of information and their applications*. Wiley, 1994.

[10] R. Linsker. Self-organization in a perceptual network. *IEEE Computer*, 21(3):105–117, 1988.

[11] C. Liu and H. Shum. Kullback-Leibler boosting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Madison, WI, 2003.

[12] S. Lyu. Infomax boosting. Technical Report TR2004-511, Department of Computer Science, Dartmouth College, 2004.

[13] R. Meir and G. Rätsch. An introduction to boosting and leveraging. In S. Mendelson and A. Smola, editors, *Advanced Lectures on Machine Learning, LNCS*, pages 119–184. Springer, 2003.

[14] E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Statistics*, 33:1,065–1,076, 1962.

[15] P. Phillips, H. Moon, P. Rauss, and S. Rizvi. The FERET evaluation methodology for face recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10), 2000.

[16] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Annual Conference on Computational Learning Theory (COLT)*, 1998.

[17] T. Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research*, 3:1415–1438, 2003.

[18] N. Vasconcelos. Feature selection by maximum marginal diversity: optimality and implications for visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[19] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[20] M. Yang, N. Ahuja, and D. Kriegman. Detecting faces in images: a survey. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.