A Methodology for Information Theoretic Feature Extraction

John W. Fisher III
MIT Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
fisher@ai.mit.edu

José C. Principe
Computational NeuroEngineering Laboratory
University of Florida
Gainesville, FL 32611
principe@cnel.ufl.edu

Abstract

We discuss an unsupervised feature extraction method which is driven by an information theoretic based criterion: mutual information. While information theoretic signal processing has been examined by many authors the method presented here is more closely related to the approaches of Linsker (1988,1990), Bell and Sejnowski (1995), and Viola et al (1996). The method we discuss differs from previous work in several aspects. It is extensible to a feed-forward multi-layer perceptron with an arbitrary number of layers. No assumptions are made about the underlying PDF of the input space. It exploits a property of entropy coupled with a saturating nonlinearity resulting in a method for entropy manipulation with computational complexity proportional to the number of data samples squared. This represents a significant computational savings over previous methods (Viola et al, 1996). As mutual information is a function of two entropy terms, the method for entropy manipulation can be directly applied to the mutual information as well.

1. Introduction

Classification is often hindered by the so called "curse of dimensionality". It is often the case that the dimensionality of the observed signal is too large to reliably/robustly construct a classifier. Consequently, various methods have been applied in order to reduce the dimensionality. This process, referred to as feature extraction, often results in improved performance of a nonparametric classifier.

It is imperative, however, that the driving criterion of any feature extraction method somehow be related to the overall system objective; namely classification. Suitable feature extraction criteria for classification are not always easily applied (e.g. likelihood ratios which require prior knowledge of the underlying probability density function). As a result, sub-optimal feature sets or user defined ad hoc features based on intuitive assumptions (without rigorous relationship to classification) are used for classification.

We have recently presented a method which derives features which are relevant for classification [4, 5, 6]. The method applies Linsker's *Principle of Information Maximization*, which seeks to transfer maximum *information* about a signal from the input to the output of a mapping, as the criterion for feature extraction [7]. As a result, we seek parameters of a general (differentiable) nonlinear mapping such that the mutual information between the observed output and the class of interest is maximized. Mathematically, mutual information is formulated as

$$I(C,y) = h_Y(y) - h_{Y|C}(y|C) \qquad (1)$$

where $I(C,y)$ is the mutual information of the RVs C and Y, and $h([\,])$ is the differential entropy measure [9]. Given the random vector (RV), $Y \in \Re^N$, the differential entropy is defined as

$$h_Y(u) = -\int_{-\infty}^{\infty} \log(f_Y(u)) f_Y(u) du \;, \qquad (2)$$

where $f_Y(u)$ is the probability density function of the RV, the base of the logarithm is arbitrary, and the integral is N-fold.

Previous authors [1, 11] have proposed similar techniques with application to various problems (e.g. blind source separation, pose estimation, etc.), but none have addressed feature extraction in the context of information theoretic processing (the method proposed here being also more general). In contrast to Bell and Sejnowski [1] the method here can be generalized to a multi-layer perceptron with an arbitrary number of hidden layers and nodes. Furthermore, the application is not specific to uni-modal distributions. In contrast to the method of Viola *et al* [11] we use an indirect measure of entropy rather than a direct estimate in order to determine our mapping. As a result, the optimization of entropy can be modeled as local interactions between the observed data samples in the output space.

In this discussion we address three primary issues:

1. The appropriateness of mutual information as a criterion for feature extraction in the context of classification.

0-7803-4859-1/98 $10.00©1998 IEEE

2. Significant computational reduction as compared to our previous algorithms.
3. The perspective of mutual information as local interaction of the data in the output space.

2. Mutual Information And Classification

The use of mutual information for classification can be motivated by Fano's inequality [3] which gives a lower bound for the probability of error (or conversely an upper bound on the probability of correct classification) when estimating a discrete RV from another RV as a function of mutual information. Fano's inequality is stated as follows, given the discretely distributed RV C (representing the class) and a related RV Y, the probability of incorrectly estimating C based on an estimate derived from observations of Y is lower bounded by

$$P(C \neq \hat{C}) \geq \frac{h(C|Y)-1}{\log(N)} = \frac{h(C)-I(C,Y)-1}{\log(N)} \qquad (3)$$

where N is the number of classes represented by the RV C and $\hat{C}$ is the estimate of C after observing Y. We see from equation 3, that the lower bound on the error probability is minimized when the mutual information between C and Y is maximized.

Fano's inequality, therefore, justifies the use of mutual information as a feature extraction criterion for classification. The approach is depicted in figure 1 with regards to a Bayesian framework. The probability density function of the high-dimensional observation X is conditioned on the class, C. The feature vector, $Y = g(X, \alpha)$, is derived from the observation of X and is itself a random vector prior to observation. It is from the feature vector, Y, that we wish to estimate the class. Our goal is to choose the parameters, α, of the mapping $g([\ ], \alpha)$ such that the mutual information between Y and C is maximized. We are still left with the task of determining the estimator, $\hat{C}$, however, from Fano's inequality we know that if $I(C, Y)$ is maximized, the lower bound on the classification error will be minimized.

3. Simplification Of The Learning Algorithm

Having motivated mutual information as an optimization criterion, we must still determine the mapping parameters, α. Examination of equation 1 shows that the mutual information, $I(C, Y)$, can be written as a function of two entropy terms, $h(Y)$ and $h(Y|C)$. In the context of classification, $h(Y)$ represents the entropy of the observations over all classes, while $h(Y|C)$ represents the class conditional entropy. An algorithm which manipulates entropy (independently applied) is sufficient for maximizing mutual information. We have presented such an algorithm

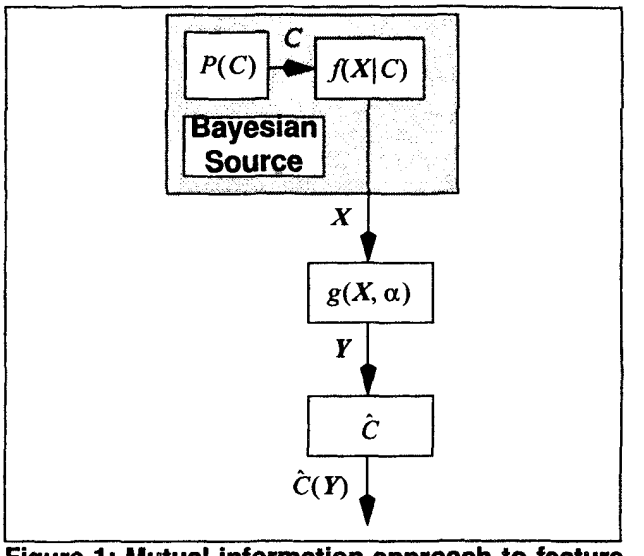

Figure 1: Mutual information approach to feature extraction. An observation of the random variable X is generated by the probability density function $f(X|C)$ which is conditioned on the discrete random variable C which is characterized by the discrete probability density function $P(C)$. The features, Y, are used to estimate C.

in previous work [4, 5, 6] which resulted in the adaptation scheme depicted in figure 2.

The method exploits the following property of differential entropy:

> If the RV is restricted to a finite range in $\Re^N$, differential entropy is maximized for the *uniform* distribution.

Our algorithm, therefore, seeks parameters, α, such that the distribution observed at the point, y_i, is as close to (maximizing entropy) or distant from (minimizing entropy) the uniform distribution as possible. This approach to entropy manipulation is similar to that of Bell and Sejnowski [1]. There are, however, several differences. The algorithm discussed here works equally well for multi-modal distributions as well as uni-modal distributions. More significantly, there are no restrictions placed on the number of hidden layers or nodes in the multi-layer perceptron structure used as the mapping.

Towards the goal described above we require a distance metric and a differentiable estimator of the output distribution. The estimator we use is the Parzen window method [10]. The Parzen window estimate of the probability distribution, $f_Y(u)$, of a random vector $Y \in \Re^N$ at a point u is defined as

$$f_Y(u) = \left(\frac{1}{N_y}\right) \sum_{i=1}^{N_y} \kappa(y_i - u) \; . \qquad (4)$$

The vectors $y_i \in \Re^N$ are observations of the random vector and $\kappa([\ ])$ is a kernel function. We choose the

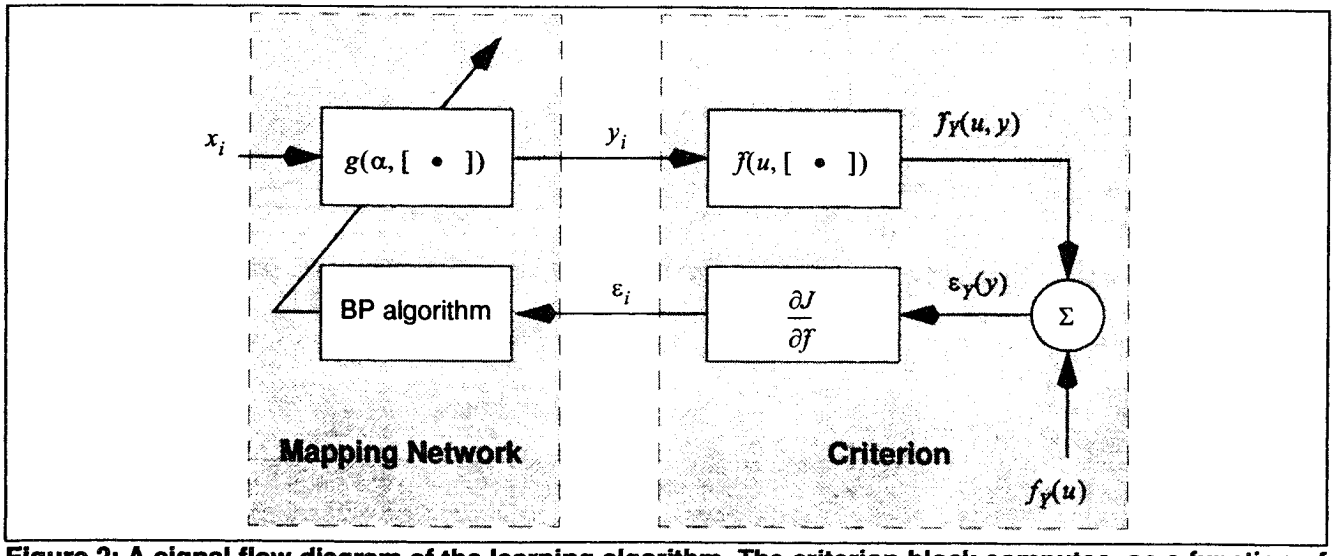

Figure 2: A signal flow diagram of the learning algorithm. The criterion block computes, as a function of the observed outputs, the error direction for the mapping network.

gaussian kernel since we require that $\kappa([\ ])$ be differentiable everywhere.

As our distance criterion we use the integrated squared error (ISE) between the observed output distribution, $f_Y(u, y)$ at a point u over a set of observations y, and the uniform distribution, $f_Y(u)$.

$$J = \int_{\Omega_Y} (f_Y(u) - f_Y(u, \{y\}))^2 du = \frac{1}{2} \int_{\Omega_Y} (f_Y(u) - f_Y(u, g(\{x\}, \alpha)))^2 du \quad (5)$$

It is this choice of criterion coupled with the multi-layer perceptron architecture which leads to a computationally simple algorithm for manipulating entropy. Viola *et al* [11] also use the Parzen window estimator for entropy manipulation enabling a means by which to apply gradient descent to the mapping parameters. A significant difference, however, is that the method of Viola *et al* estimates entropy directly. By contrast, the method here uses an indirect measure of entropy (ISE) coupled with a saturating nonlinearity. This approach, as we will see, enables entropy manipulation to be modeled as local interaction between observations in the output space.

In our previous work [4, 5, 6] the straightforward evaluation of the gradient of the criterion with respect to the mapping parameters, α, led to the following update rule for manipulating the entropy of the mapping,

$$\Delta\alpha \propto \left(\frac{1}{N_y}\sum_i (\varepsilon_Y(u, \{y\}) * \kappa'(u))\Big|_{u = y_i}\right)\frac{\partial}{\partial\alpha} g(\alpha, x_i) = \left(\frac{1}{N_y}\sum_i \varepsilon_i\right)\frac{\partial}{\partial\alpha} g(\alpha, x_i) \quad (6)$$

where N_Y is the number of training exemplars, $\varepsilon_Y(u, \{y\})$ is the computed error distribution between the estimated output distribution and the desired uniform distribution, $\kappa'(u)$ is the gradient of the estimator kernel, and $\partial g(\alpha, x_i)/\partial\alpha$ are the mapping sensitivities (which we compute using error back-propagation). Excluding the mapping sensitivities, the remaining terms can be combined to compute an error direction term, ε_i, associated with each training sample. The error direction term is the convolution of the observed error distribution with the kernel gradient. A more general discussion of this approach is presented in Fisher and Principe [4, 5, 6].

The straightforward approach has one significant drawback in that the algorithm, as implemented, requires the evaluation of the convolution term (and the Parzen window estimate) at a sufficient number of points in the output space. Consequently the computational complexity of the algorithm implement in this way is proportional to

$$O(N_y^{N_d+2}), \quad (7)$$

where N_d is the dimension of the output space.

Fortunately, the dimensionality of the output space is controlled by the designer. Equation 7, however, poses a fundamental computational limitation to the dimensionality of the subspace mapping. This limitation, however, is only valid if the implicit error term is computed in the straightforward manner that the equations imply. Further examination of the gradient of the ISE criterion results in significant reduction in the computational complexity.

Expanding the error direction term, ε_i, yields

$$\begin{aligned}\varepsilon_i &= \varepsilon_Y(u, \{y\})*\kappa'(u)\big|_{u=y_i} \\ &= (f_Y(u) - f_Y(u, \{y\}))*\kappa'(u)\big|_{u=y_i} \\ &= (f_Y(u) - y(u)*\kappa(u))*\kappa'(u)\big|_{u=y_i} \\ &= (f_Y(u)*\kappa'(u)) - y(u)*\kappa(u)*\kappa'(u)\big|_{u=y_i} \\ &= f_r(u) - y(u)*\kappa_a(u)\big|_{u=y_i} \\ &= f_r(y_i) - \sum_{j\neq i} \kappa_a(y_i - y_j)\end{aligned} \quad (8)$$

where, $y(u)$, representing the location of the training samples in the output space, is written as

$$y(u) = \sum_{i=1}^{N_Y} \delta(u - y_i).$$

The terms $\kappa_a(u)$ and $f_r(u)$ are termed the attractor kernel and the topology regulating term, respectively. Equation 8 overcomes the fundamental limitation implied by equation 7. Both terms can be computed analytically (for the gaussian kernel and the uniform distribution). More importantly, the computational complexity of equation 8 is only of order N_y for each y_i. Substituting 8 into the update rule of equation 6 results in a computational complexity which is quadratic in the number of exemplars, simplifying to

$$O(N_y^{N_d+2}) \rightarrow O(N_y^2). \quad (9)$$

Of particular interest are the forms of $\kappa_a(u)$ and $f_r(u)$. The analytic forms for the gaussian kernel (with diagonal covariance) and the uniform distribution are

$$\begin{aligned}\kappa_a(u) &= \kappa(u)*\kappa'(u) \\ &= -\left(\frac{1}{2^{N+1}\pi^{N/2}\sigma^{N+2}}\right)\exp\left(\frac{1}{4\sigma^2}(u^T u)\right)u\end{aligned} \quad (10)$$

$$f_r(u) = \begin{bmatrix} \frac{1}{a^N}\left(\prod_{i\neq 1}\frac{1}{2}\left(\operatorname{erf}\left(\frac{u_i+\frac{a}{2}}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{u_i-\frac{a}{2}}{\sqrt{2}\sigma}\right)\right)\right) \\ \times\left(\kappa_1\left(u_1+\frac{a}{2},\sigma\right) - \kappa_1\left(u_1-\frac{a}{2},\sigma\right)\right) \\ \vdots \\ \frac{1}{a^N}\prod_{i\neq N}\frac{1}{2}\left(\operatorname{erf}\left(\frac{u_i+\frac{a}{2}}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{u_i-\frac{a}{2}}{\sqrt{2}\sigma}\right)\right) \\ \times\left(\kappa_1\left(u_N+\frac{a}{2},\sigma\right) - \kappa_1\left(u_N-\frac{a}{2},\sigma\right)\right) \end{bmatrix}. \quad (11)$$

where N is the dimension of the kernel, σ is the size of the kernel, a is the extent in all dimensions of the uniform distribution, and $\kappa_1(u, \sigma)$ indicates the one dimensional gaussian kernel evaluated at u with standard deviation σ. These functions are shown for the two dimensional case in figures 3 and 4. From the figure we can see that $\kappa_a([\ ])$ represents the influence that each observation has on its local surrounding, while $f_r([\ ])$ represents the influence on each sample near the boundary of the region of support.

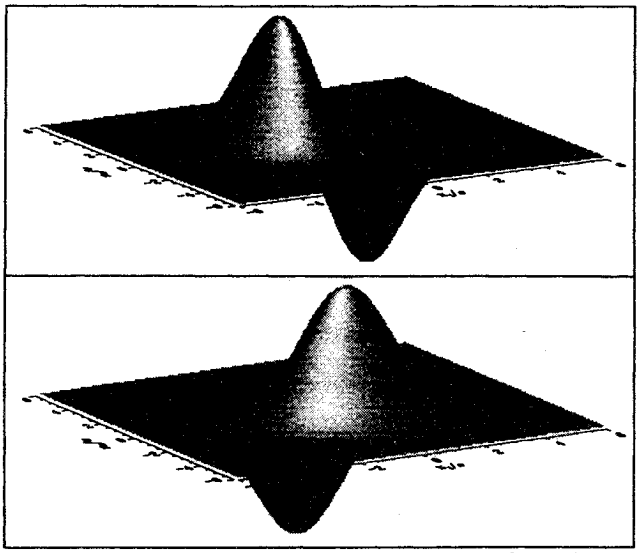

Figure 3: Two dimensional attractor functions. The x_1-component is shown at the top while the x_2-component is shown at the bottom. The function represents the local influence of each data point in the output space.

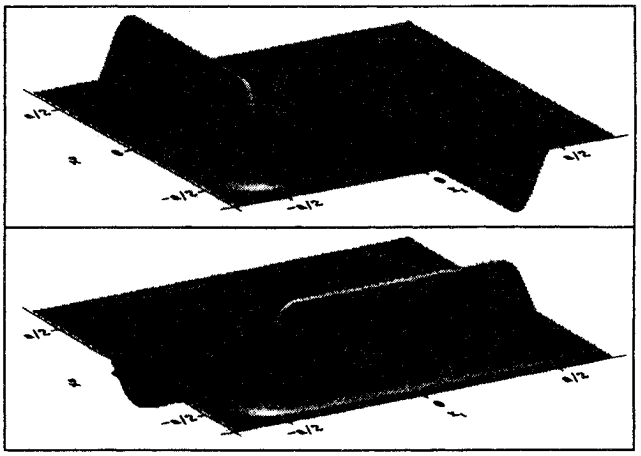

Figure 4: Two dimensional regulating function. The x_1-component is shown at the top while the x_2-component is shown at the bottom,.

From this perspective, we see that entropy (mutual information) can be modeled as a local attraction/repulsion

between samples in the output space. As we maximize entropy the samples repel each other and as we minimize entropy (as in the conditional entropy term of mutual information) the samples attract each other. From a classification standpoint, this would be a desirable property. Samples from the same class would map to compact locations, while classes from separate classes will repel.

4. Conclusions

We have presented a general method for manipulating entropy of a mapping and developed an algorithm of sufficiently low complexity that makes it practical. The method is extensible to general, differentiable nonlinear mappings. We have already shown that it fits easily into the backpropagation formalism [4, 5, 6]. The method is not constrained by assumptions about the underlying distribution at the input of the mapping. In addition, it is not limited to unimodal distributions as in the case of Bell and Sejnowski [1]. Furthermore, the computational complexity is independent of the dimension at the output space, although the quality and generality of the features, as in any nonparametric approach, will be related to the number of training samples.

During the discussion we demonstrated the appropriateness of mutual information in the context of classification. We also demonstrated that the computational complexity of our previous results could be greatly simplified, yielding an algorithm which is quadratic in the number of training samples. This simplification also led to a perspective by which entropy manipulation can be modeled as local interactions among the training samples in the output space. This perspective is important as it leads to even more simplification of the computational complexity of the algorithm, which we will be reporting on in the future.

Acknowledgments

This work was partially supported by DARPA grant F-33615-97-1-1019.

References

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

[2] Deco, G., and D. Obradovic (1996); *An Information-Theoretic Approach to Neural Computing*, Springer-Verlag, New York.

[3] Fano, R. M. (1961); *Transmission of Information: A Statistical Theory of Communication*, Wiley, NY.

[4] Fisher J., and Principe, J. C. (1995); "Unsupervised learning for nonlinear synthetic discriminant functions", *Proceedings of SPIE*, 2752: 1-13.

[5] Fisher J., and Principe, J. C. (1997a); "A Nonparametric Method for Information Theoretic Feature Extraction", *Proceedings of the DARPA Image Understanding Workshop*, New Orleans, LA, 1997.

[6] Fisher J., and Principe, J. C. (1997b); "Entropy Manipulation of Arbitrary Nonlinear Mappings", to appear in the *Proceedings of the Neural Networks for Signal Processing Workshop*, Amelia Island, FL, 1997.

[7] Linsker, R. (1988); "Self-organization in a perceptual system", *Computer*, 21: 105-117.

[8] Linsker, R. (1990); "How to generate ordered maps by maximizing the mutual information between input and output signals", *Neural Computation*, 1:402-411.

[9] Papoulis, A. (1991), *Probability, Random Variables, and Stochastic Processes* (3rd ed.), McGraw-Hill, New York.

[10] Parzen, E. (1962); "On the estimation of a probability density function and the mode", *Ann. Math. Stat.* 33: 1065-1076.

[11] Viola, P., N. Schraudolph, and T. Sejnowski,(1996); "Empirical entropy manipulation for real-world problems", *Neural Information Processing Systems* 8, to appear in published proceedings.