DO HEBBIAN SYNAPSES ESTIMATE ENTROPY?

Deniz Erdogmus, Jose C. Principe, Kenneth E. Hild II

CNEL, Electrical & Computer Engineering Dept.
University of Florida, Gainesville, FL 32611.
[deniz,principe,hildk]@cnel.ufl.edu

Abstract. Hebbian learning is one of the mainstays of biologically inspired neural processing. Hebb's rule is biologically plausible, and it has been extensively utilized in both computational neuroscience and in unsupervised training of neural systems. In these fields, Hebbian learning became synonymous for correlation learning. But it is known that correlation is a second order statistic of the data, so it is sub-optimal when the goal is to extract as much information as possible from the sensory data stream. In this paper, we demonstrate how *information* learning can be implemented using Hebb's rule. Thus the paper brings a new understanding to how neural systems could, through Hebb's rule, extract information theoretic quantities rather than merely correlation.

INTRODUCTION

Hebb's rule states: "When an axon of cell A is near enough to excite cell B or repeatedly or consistently takes part in firing it, some growth or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased" [1]. This principle has been translated in the neural networks literature as: In Hebbian learning, the weight connecting a neuron to another is incremented proportional to the product of the input to the neuron and its output [2]. This formulation then can be shown to maximize the correlation between the input and the output of the neuron whose weights are updated through the use of Widrow's stochastic gradient on a correlation-based cost function [2,3]. Correlation has been declared the basis of learning, and Hebb's law has mostly shaped our understanding of the operation of the brain and the process of learning. However, we know that correlation describes simple second order statistics of the data, and as such it is sub-optimal when the goal is to process information. Our brains must exploit much more than simply correlation from the neural activity in order to achieve their level of performance in information processing. From an abstract perspective, the learning-from-examples scenario starts with a data set, which globally conveys information about a real world event, and the goal is to capture the information in the weights of a learning machine. Information Theory (IT) should be the mathematical infrastructure used to quantify this change in state because it is the best possible approach to deal with manipulation of information [4]. Shannon, in a 1948 classical paper, laid down the foundations of IT [5]. IT has had a tremendous impact in the design of efficient and reliable communication

systems [6,7] because it is able to answer two key questions: what is the best possible (minimal) code for our data, and what is the maximal amount of information which can be transferred through a particular channel. In spite of its practical origins, IT is a deep mathematical theory concerned with the very essence of the communication process. IT has also impacted statistical mechanics by providing a clearer understanding of the nature of entropy, as was illustrated by Jaynes [8].

Due to the emphasis on principles, information theory has also played a role in explaining biological information processing. Barlow enunciated the principle of redundancy reduction [9] building on earlier work of Attneave on visual perception [10] and utilized it to train (unsupervised learning) the weights of a linear neural network. He also proposed factorial or minimum entropy coding [11]. The development of sparse codes has been recently advanced by many researchers [12,13]. Atick [14] and Atick and Redlich [15] demonstrated that feature extraction could be accomplished from noisy inputs by maximizing mutual information between the input and the output, which is a form of redundancy reduction. The recent interest in sparse representations has been triggered by Fields [16], and a subsequent paper in Nature, which demonstrated emergence of simple cell receptive fields when learning a sparse code for natural images [13].

Probably the most insightful application of information theory to brain theory is Linsker's principle of maximum information preservation, also called the Infomax principle [17]. Linsker enunciated a self-organizing principle for neuronal assemblies based on the simple idea that the role of a neuronal assembly is to transfer to its output as much information as possible about its input. He formulated this principle in analogy to the channel capacity theorem as the maximization of the mutual information between the input and output. This is a profound idea that shines new light on the organizational principles of the brain.

One of the difficulties of all these studies is that there is an abyss between the sophisticated computation required to estimate Shannon's entropy and mutual information and the reality of the local computation of the Hebbian synapse. We will show in this paper, however, that if a kernel-based nonparametric estimator for Shannon's entropy is utilized, then the Hebbian synapse can in fact estimate entropy over time instead of simply estimating correlation. We also demonstrate that the same goal could also be reached through the unconventional Renyi's entropy.

The structure of this paper is the following. First we briefly explain and provide references of our nonparametric approach to estimate entropy directly from samples for both Shannon's and Renyi's entropy definitions. Then we derive the stochastic information gradient (SIG) to adapt the parameters of a linear, or nonlinear system, to maximize or minimize entropy [18]. While exploring the mathematical properties of the SIG algorithm, we will establish the similarity between a special case of the SIG algorithm and Hebbian and anti-Hebbian terms, which raised the question of whether information processing is possible through Hebbian learning. Finally we present a simple example to demonstrate the convergence of the algorithm to the desired solutions.

SIG ALGORITHM FOR ADALINE

In order to arrive at the SIG algorithm, we start with the derivation of the nonparametric entropy estimator that employs Parzen windowing. First consider Shannon's entropy, which is given by

$$H_S(y) = E_y[-\log f_y(y)] \tag{1}$$

Suppose we utilize the stochastic value of this quantity at time k, such that the expectation is dropped and the argument of this operation is evaluated at the most recent sample.

$$H_S(y) \approx -\log f_y(y_k) \tag{2}$$

Since in practice the analytical pdf of the random variable y is not available, we employ the nonparametric Parzen window estimator over the last L samples of the signal [21].

$$\hat{H}_S(y) = -\log \frac{1}{L} \sum_{j=k-L}^{k-1} \kappa_\sigma(y_k - y_j) \tag{3}$$

Assuming that the samples of y are generated by an ADALINE structure, i.e. $y_k = w^T x_k$, where x_k is the input vector and w is the weight vector, the stochastic gradient estimate for Shannon's entropy is found as (in terms of a column vector)

$$\frac{\partial \hat{H}_S(y)}{\partial w} = -\frac{\sum_{j=k-L}^{k-1} \kappa'_\sigma(y_k - y_j)(x_k - x_j)}{\sum_{j=k-L}^{k-1} \kappa_\sigma(y_k - y_j)} \tag{4}$$

This expression is called the stochastic information gradient (SIG) [18]. SIG can also be derived using Renyi's entropy, a parametric family of functions. For a random variable y it is defined as [19]

$$H_\alpha(y) = \frac{1}{1-\alpha} \log E_y[f_y^{\alpha-1}(y)] \tag{5}$$

We named the argument of the log in (5) as *the information potential*, not arbitrarily, but because it shares the properties of physical potential fields, when the formulation is complete [20]. The expectation in the information potential could be dropped to obtain a stochastic estimate of this quantity. Notice that minimization or maximization of entropy is equivalent to minimization or maximization of the information potential, depending on the value of α. Using Parzen windowing over the last L samples, at time k, we obtain the stochastic quantity

$$\hat{V}_\alpha(y) = \left(\frac{1}{L} \sum_{j=k-L}^{k-1} \kappa_\sigma(y_k - y_j) \right)^{\alpha-1} \tag{6}$$

Then, for an ADALINE, the stochastic gradient of Renyi's entropy becomes

$$\frac{\partial \hat{H}_\alpha(y)}{\partial w} = -\frac{\partial \hat{V}_\alpha(y)/\partial w}{\hat{V}_\alpha(y)} = -\frac{\sum_{j=k-L}^{k-1} \kappa'_\sigma(y_k - y_j)(x_k - x_j)}{\sum_{j=k-L}^{k-1} \kappa_\sigma(y_k - y_j)} \tag{7}$$

which is identical to (3). In (3) and (7), $\kappa_\sigma(.)$ is called the kernel function of Parzen windowing, usually selected to be a symmetric, continuous and differentiable pdf, whose width is determined by the parameter σ. In [22], we have established the link between the kernel function and the convolution smoothing for global optimization, and we also demonstrated the link between the minimum error-entropy (MEE) criterion and minimum mean-square-error (MMSE) criterion in supervised learning.

Note that, when maximizing or minimizing the output entropy of ADALINE, just like in Oja's rule [23], we would need to normalize the weights to keep them from growing without bound or decaying to zero.

Alternative SIG expressions for Shannon's and Renyi's definitions of entropy could also be obtained by simply utilizing a single sample from the past in the Parzen pdf estimate and using the window of L samples to approximate the expectation value operator with a sample mean. As a result, assuming an ADALINE structure, we obtain the following alternative SIG expressions for Shannon's and Renyi's entropies, respectively.

$$\frac{\partial \hat{H}_S(y)}{\partial w} = -\frac{1}{L} \sum_{j=k-L+1}^{k} \frac{\kappa'_\sigma(y_j - y_{j-1}) \cdot (x_j - x_{j-1})}{\kappa_\sigma(y_j - y_{j-1})} \tag{8}$$

$$\frac{\partial \hat{H}_\alpha(y)}{\partial w} = -\frac{\sum_{j=k-L+1}^{k} \left[\kappa_\sigma^{\alpha-2}(y_j - y_{j-1}) \cdot \kappa'_\sigma(y_j - y_{j-1}) \cdot (x_j - x_{j-1}) \right]}{\sum_{j=k-L+1}^{k} \kappa_\sigma^{\alpha-1}(y_j - y_{j-1})} \tag{9}$$

SIG given in (3) is, in effect, very similar to Widrow's stochastic gradient in LMS; we assume only the instantaneous increment in the output samples, i.e. $y_k - y_{k-1}$, is available at the k^{th} iteration of adaptation. Moreover, it is trivial to see that the expected value of this gradient is the actual gradient of the Shannon's entropy for the output pdf estimated with Parzen windowing. For a small kernel size and a large number of samples, this estimate is very close to the true pdf. An interesting special case of (3), (8), and (9) is for L=1.

$$\frac{\partial \overline{H}_\alpha(y_k)}{\partial w} = -\frac{\cdot \kappa'_\sigma(y_k - y_{k-1}) \cdot (x_k - x_{k-1})}{\kappa_\sigma(y_k - y_{k-1})} \tag{10}$$

The convergence properties of this algorithm for entropy minimization was studied in detail in [24]. Extensions to differentiable (with respect to their weights) nonlinear systems are also possible [18].

RELATIONSHIP BETWEEN SIG AND HEBB'S RULE

In general, we prefer using differentiable and symmetric kernels in Parzen windowing; differentiability is required to guarantee proper evaluation of the gradient in adaptation, and symmetry is preferred to prevent biasing the mean of the estimated pdf. Suppose Gaussian kernels are employed. In that case, σ naturally becomes the standard deviation of the kernel, and we have the following.

$$\kappa'_\sigma(x) = \frac{-x}{\sigma^2}\kappa_\sigma(x) \tag{11}$$

Substituting (11) in (10), we obtain the explicit expression

$$\frac{\partial \overline{H}_\alpha(y_k)}{\partial w} = \frac{1}{\sigma^2}(y_k - y_{k-1}) \cdot (x_k - x_{k-1}) \tag{12}$$

We clearly see from (12) that employing Hebbian and anti-Hebbian terms involving the two most recent values of the input and the output results in a stochastic estimate of the information gradient. When interpreted under the viewpoint of classical Hebbian learning, (12) states that it is possible to maximize the entropy by applying the Hebbian rule to the instantaneous increments in the input and the output signals. Thus, the neuron implements information learning rather than merely implementing correlation learning.

Now, consider the general case where any differentiable symmetric kernel function may be used in Parzen windowing. Since only the Gaussian distribution satisfies the differential equation in (11), it is the only kernel choice that results in this special case given by (12), which reduces to a learning algorithm that is Hebbian (on the increments) in the classical sense. In general, since we use symmetric and differentiable kernels that are pdfs themselves, we get the following update rule

$$\frac{\partial \overline{H}_\alpha(y_k)}{\partial w} = f(y_k - y_{k-1}) \cdot (x_k - x_{k-1}) \tag{13}$$

where $f(x) = -\kappa'_\sigma(x)/\kappa_\sigma(x)$, and satisfies the condition $sign(f(x)) = sign(x)$. Thus, the update amount that would be applied to the weight vector is still in the same direction as would be in the classical Hebbian learning; however, it is scaled nonlinearly depending on the value of the increment that occurred in the output of the neuron. This is in fact consistent with Hebb's principle, and is a good example that demonstrates the product of the output and the input signals is not the only possibility for the weight update to implement Hebb's principle.

CASE STUDIES

In this section, we present two case studies in which an ADALINE is trained to yield maximum entropy at the output subject to the unit-norm constraint on the weight vector. The weights are updated using SIG. In both examples, training is carried out using two different kernel function choices: Gaussian and Cauchy. Explicitly, these kernel functions are given by

$$G_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad , \quad C_\sigma(x) = \frac{1}{\pi \cdot \sigma} \cdot \frac{1}{1+(x/\sigma)^2} \tag{14}$$

The weights are normalized to unit-length after each update, and the information about the previous output values are modified using this normalized weight vector. In the first example, 100 samples from a 2-dimensional joint Gaussian distribution are generated and the ADALINE is trained to maximize the entropy at the output. The step sizes are chosen as 10^{-5}, and 10^{-3}, for Gaussian and Cauchy kernels, respectively, and both kernel sizes are chosen as 0.1. Fig. 1a shows the samples and the directions deduced. Fig. 1b shows the convergence of the weight vector angle to the ideal solution, which corresponds to the 1st principal component in this case, where the data distribution is Gaussian [20]. The choice of kernel size and step size are important issues that affect convergence time and misadjustment.

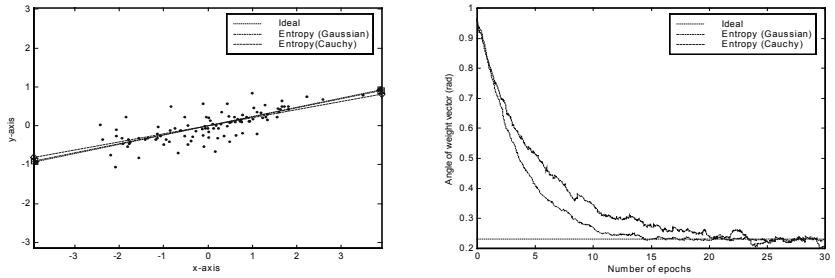

Figure 1: a) Data samples and the derived directions b) Convergence of the angles

In the second example 50 samples of a 2-dimensional random vector are generated such that the x-coordinate is uniform, y-coordinate is Gaussian, and the (sample) covariance matrix is identity. In this case, PCA is unable to deduce any maximal variance direction since the variance along each direction is the same. On the other hand, using the maximum (minimum) entropy approach, we can extract the direction along which the data exhibits the most (least) uncertainty. Fig. 2a shows the estimated entropy vs. direction of weight vector for both kernels, and Fig. 2b shows the convergence of weights from 5 different initial conditions.

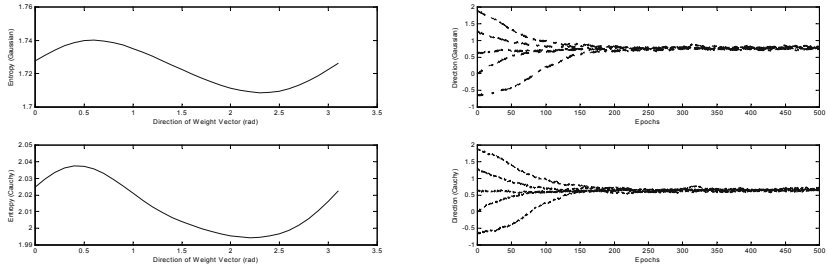

Figure 2: a) Entropy vs. direction b) Convergence from different initial conditions

We remark that, in this example, as the number of samples increases, the estimated distributions will converge to the actual distributions (for small kernel sizes), which are between uniform and Gaussian. Hence, the asymptotically ideal solution for the angle will be $\pi/2$, since the Gaussian distribution has the largest entropy among distributions of fixed variance [7].

The third case study we present here is a Monte Carlo analysis of the misadjustment of the stochastic gradient given in (12). In each *run* of this analysis, we use 10000 samples from a 2-dimensional jointly Gaussian random vector, as was done in the first example. The eigenspread ($\lambda_{max}/\lambda_{min}$) of the covariance matrix is one of the controlled variables. The kernel size σ is scaled up or down with the standard deviation of the maximum entropy projection (this could be estimated from the samples) as $\sigma = \sqrt{\lambda_{max}}\,\sigma_0$, where σ_0 is a base-value for the kernel function selected for unit-variance samples. This latter variable is also part of the controlled parameters of the Monte Carlo simulations. Specifically, we control the ratio η/σ_0^2, where η is the learning rate of the steepest ascent algorithm.

We perform these series of Monte Carlo simulations varying η/σ_0^2 from 10^{-3} to 10^{-2} and varying $\sqrt{\lambda_{max}/\lambda_{min}}$ from 2 to 10. For each pairwise combination of these parameters, 10 experiments were performed, starting from random initial conditions and using the last 3000 samples to compute the MSE between the estimated maximum entropy direction and the actual solution found using the unbiased sample covariance estimate for the whole 10000-sample training set. The MSE values obtained over the 10 runs are then averaged to produce the results summary depicted in Fig. 3. These results are presented in terms of root-mean-square (RMS) values of the error (in degrees) between the estimated direction and the true direction (notice the log-scale of the RMS axes in all three subplots). Both cross-section plots and the 2-argument 1-output mesh plot of the RMS Error versus learning rate and eigenspread are provided for convenience.

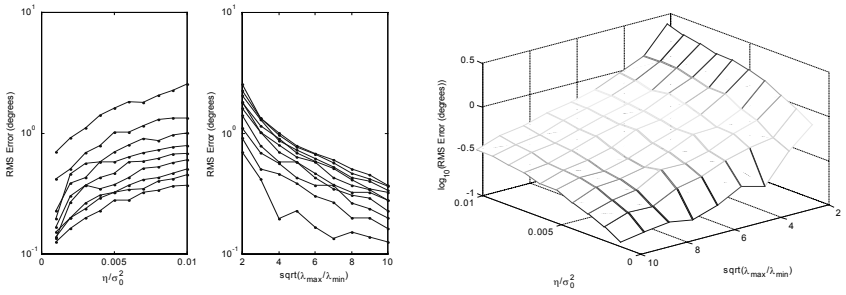

Figure 3: a) RMS Error (in degrees) versus η/σ_0^2 for different eigenspread values

b) RMS Error (in degrees) versus $\sqrt{\lambda_{max}/\lambda_{min}}$ for different learning rates

c) RMS Error (in degrees) versus η/σ_0^2 and $\sqrt{\lambda_{max}/\lambda_{min}}$

From the results shown in Fig. 3, we observe that the misadjustment increases as the learning rate increases and as the eigenspread decreases, as expected. The affect of learning rate is intuitively obvious from our knowledge on the effect of step size on the misadjustment of the well-known LMS algorithm. The intuition behind the observed effect of the eigenspread is that with increasing eigenspread, the entropy of the projection to the desired direction increases causing the differences between consecutive samples of the input vector and the output value to become larger, thus emphasizing the distinction between the maximum-entropy direction and all the other directions. Therefore, it becomes easier for the algorithm to determine and converge to the optimal solution sought.

CONCLUSIONS

In this paper we presented a nonparametric entropy estimator, whose stochastic gradient led to a local, sample-wise expression that involves the product of a nonlinear function of the differences between consecutive output samples and the difference between the corresponding input samples. The derived stochastic entropy gradient expressions (all three of them) can be utilized in the *information theoretic* solution of many engineering problems where training of adaptive systems according to information theoretic criteria is required. These *stochastic information gradient* (SIG) expressions allow the designer to manipulate the information at the output of a learning system on a sample-by-sample basis; therefore, they are extremely useful for fast, on-line *information theoretic learning*.

For the special case of single-sample windows, this stochastic gradient defaulted to a combination of Hebbian and anti-Hebbian learning, among pairs of consecutive samples, acting on their instantaneous value increments in value instead of their instantaneous values, as in the classical sense. We have investigated the ability of this simplified SIG algorithm to determine maximum-entropy directions in different sets of random variables and demonstrated that it successfully forces the linear adaptive system to converge to the desired solution. A Monte Carlo analysis of the effects of the learning rate and the eigenspread of the data on the misadjustment of the algorithm in determining the maximum-entropy direction was also conducted. This analysis verified the expectation that the misadjustment increases with increasing learning rate and decreasing eigenspread.

There are two main conclusions from this study that we would like to address. This relationship between entropy and Hebbian update principle encourages us to study the realism of this learning model in neuronal interactions in the brain; experiments could be set up to see if, in fact, synaptic strength is only dependent upon the level of excitability of the neuron or if, as the formula predicts, the efficacy of the synapse is also modulated by the action potential's intervals (verbal communication with several neuroscience experts revealed the possibility of a similar process in neuron synapses). If our prediction is correct, then we have a very gratifying result that increases even further our admiration for biological

processes. In fact, Hebbian learning could adapt the synapses with information gradients, and therefore neural assemblies will manipulate entropy, which from the point of information processing provides all that can be known about the data streams.

The second conclusion concerns the machine learning community. Even if our predictions for computational neuroscience are incorrect, Renyi's entropy estimator is a viable alternative to help us leave the "local minima" created by second order methods so pervasive in both adaptive filter theory and artificial neural network cost functions. SIG clearly shows that Hebbian type rules are estimating far more than second order statistics. It is quite extraordinary that interactions between consecutive samples are able to estimate higher order statistics. Upon employing our nonparametric estimator for entropy and seeking an on-line implementation, we began to realize how powerful local computation in space-time can really be. As a result, it is imperative that researchers move to a different paradigm, where Hebbian learning is no longer a synonym of correlation learning.

Finally, we would like to address the issue of applicability of the presented relationship to the field of information theoretic learning. In general, mutual information, not entropy alone, is employed as the adaptation criterion as it is scale invariant and exhibits other desirable properties. It is easy to write mutual information in terms of marginal and joint entropies of the signals involved, therefore a link between mutual information and Hebbian learning might as well be established following this lead. On the other hand, there are applications where a modified entropy criterion might be useful, such as blind deconvolution. By adding the *log* of the variance of the signal to its entropy, a scale invariant objective function could be constructed, which would exhibit Hebbian update rules for both entropy and the variance terms in the stochastic updates. In summary, there is still much to be researched about the links between information extraction in biological adaptive systems and their application to learning engineering systems.

Acknowledgments: This work is partially supported by the grants NSF-ECS-9900394 and ONR-N00014-01-1-0405.

REFERENCES

[1] D.O. Hebb, The Organization of Behavior: A Neuropsychological Theory, Wiley, New York, 1949.

[2] S. Haykin, Neural Networks, Prentice-Hall, New Jersey, 1999.

[3] B. Widrow, S.D. Stearns, Adaptive Signal Processing, Prentice-Hall, New Jersey, 1985.

[4] R. Fano, Transmission of Information, MIT Press, Massachussetts, 1961.

[5] C.E. Shannon, "A mathematical theory of communication," Bell Sys. Tech. J., vol. 27, pp. 379-423,623-653, 1948.

[6] R. Blahut, Principles and Practice of Information Theory, Addison Wesley, Massachussetts, 1987.
[7] T. Cover, J. Thomas, Elements of Information Theory, Wiley, New York, 1991.
[8] E. Jaynes, "Information theory and statistical mechanics," Phys. Rev., vol. 106, pp. 620-630, 1957.
[9] H. Barlow, P. Foldiak, "Adaptation and Decorrelation in the Cortex," in Computing Neuron, Durbin, Miall & Mitchison (eds.), Addison Wesley, Massachussetts, 1989.
[10] F. Attneave, "Information Aspects of Visual Perception," Psych. Rev., vol. 61, pp. 183-193, 1954.
[11] H. Barlow, T. Kaushal, G. Mitchison, "Finding Minimum Entropy Codes," Neural Computation, vol 1, no. 3, pp. 412-423, 1989.
[12] R. Zemel, P. Dayan, A. Pouget, "Probabilistic Interpretation of Population Codes," Neural Computation, vol. 10, pp. 403-430, 1998.
[13] H. Olshausen, D.J. Field, "Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images," Nature, vol. 381, pp. 607-609, 1996.
[14] J. Atick, "Could Information Theory Provide an Ecological Theory of Sensory Processing?" Network, vol. 3, pp. 213-251, 1992.
[15] J. Atick, A. Redlich, "Convergent Algorithms for Sensory Receptive Field Development," Neural Computation, vol. 5, pp. 45-60, 1993.
[16] D. Fields, "What is the Goal of Sensory Coding?" Neural Computation, vol. 6, pp. 559-601, 1994.
[17] R. Linsker, "An Application of the Principle of Maximum Information Preservation to Linear Systems," in D.S. Tourezky (ed.), Morgan-Kauffman, San Francisco, 1988.
[18] D. Erdogmus, J.C. Principe, "An On-Line Adaptation Algorithm for Adaptive System Training with Minimum Error Entropy: Stochastic Information Gradient," in Proc. of Independent Component Analysis 2001 (ICA'01), 2001.
[19] A. Renyi, Probability Theory, American Elsevier Pub. Co., New York, 1970.
[20] J.C. Principe, D. Xu, J. Fisher, "Information Theoretic Learning," in Unsupervised Adaptive Filtering, vol I, S. Haykin (ed.), Wiley, New York, 2000.
[21] E. Parzen, "On Estimation of a Probability Density Function and Mode,". in Time Series Analysis Papers, Holden-Day, Inc., California, 1967.
[22] D. Erdogmus, J.C. Principe, "Generalized Information Potential Criterion for Adaptive System Training," to appear in IEEE Trans. on Neural Networks, 2002.
[23] E. Oja, Subspace Methods for Pattern Recognition, Research Studies Press, England, 1982.
[24] D. Erdogmus, J.C. Principe, "Convergence Analysis of the Information Potential Criterion in ADALINE Training," Proc. Neural Networks in Signal Processing X, 2001.