INFORMATION THEORETIC LEARNING: RENYI'S ENTROPY AND ITS APPLICATIONS TO ADAPTIVE SYSTEM TRAINING

By

DENIZ ERDOGMUS

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2002

This work is dedicated to
all scientists and researchers,
who have lived in pursuit of knowledge,
and have dedicated themselves to the
advancement of science.

ACKNOWLEDGMENTS

I would like to start by thanking my supervisor, Dr. Jose C. Principe, for his encouraging and inspiring style that made possible the completion of this work. Without his guidance, imagination, and enthusiasm, which I admire, this dissertation would not have been possible.

I also wish to thank the members of my committee, Dr. John G. Harris, Dr. Tan F. Wong, and Dr. Mark C.K. Yang, for their valuable time and interest in serving on my supervisory committee, as well as their comments, which helped improve the quality of this dissertation.

Throughout the course my PhD research, I have been in interaction with many CNEL colleagues and I have benefited from the valuable discussions we had together about related subjects. Especially, I would like to mention Kenneth E. Hild II and Yadunandana N. Rao, whose stimulating collaboration and insightful comments improved the quality and helped greatly to the completion of this work.

Finally, I wish to express my great love for my wife, Ebru Korbek-Erdogmus. I thank Ebru for her love, caring, and support (which has been my main thrust to continue working); her patience when listening to me explain the details of my research; and for sharing my enthusiasm for what I do.

TABLE OF CONTENTS

APPENDIX

LIST OF ABBREVIATIONS

$\forall$	For all
$\Re$	Set of real numbers
$\exists$	There exists some
ADALINE	Adaptive linear neuron
AWGN	Additive white Gaussian noise
BD	Blind deconvolution
BE	Blind equalization
BER	Bit error rate
BPSK	Binary phase shift keying
BSS	Blind source separation
CDF	Cumulative density function
CMA	Constant modulus algorithm
CNEL	Computational NeuroEngineering Laboratory
ICA	Independent component analysis
IID	Independent and identically distributed
IFF	If and only if
ISI	Inter-symbol interference
ITL	Information theoretic learning
K-L	Kullback-Leibler
LMS	Least mean-square

LOG	Natural logarithm
LTI	Linear time-invariant
MEE	Minimum error entropy
MEL	Minimum energy learning
MLP	Multilayer perceptron
MMI	Minimum mutual information
MSE	Mean-square error
PCA	Principal component analysis
PDF	Probability density function
PE	Processing element
PSD	Power spectral density
QAM	Quadrature amplitude modulation
QMI-ED	Quadratic mutual information – Euclidean distance
QMI-CS	Quadratic mutual information – Cauchy-Schwartz inequality
QPSK	Quadrature phase shift keying
SIG	Stochastic information gradient
SIR	Signal-to-interference ratio
SDR	Signal-to-distortion ratio
SNR	Signal-to-noise ratio
TDNN	Time-delay neural network
WSS	Wide sense stationary

LIST OF TABLES

Table

LIST OF FIGURES

Figure

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

INFORMATION THEORETIC LEARNING: RENYI'S ENTROPY AND ITS APPLICATIONS TO ADAPTIVE SYSTEM TRAINING

By

Deniz Erdogmus

May 2002

Chairman: Dr. Jose C. Principe
Major Department: Electrical and Computer Engineering

Traditionally, second-order statistics are used as the optimality criterion in almost any adaptive system training scenario, supervised or unsupervised, with great success, thanks to the pioneering works of Wiener and Widrow. However, recently there have been new problems arising that require more than just the second-order statistics. Blind deconvolution and independent component analysis are known to necessitate higher order statistics or alternative information theoretic criteria. For a couple of years, Computational NeuroEngineering Laboratory has been working on similar problems that may be solved using information theoretic criteria. Nonparametric algorithms and associated adaptation algorithms were proposed for Renyi's quadratic entropy and approximations to Renyi's quadratic mutual information definitions. These algorithms were applied successfully to many practical problems ranging from blind source separation to feature reduction for classification.

This research aimed to investigate the full potential of information theoretic learning while filling in the mathematical gaps in previous work and providing valid and efficient simplifications to the associated learning algorithms to convey practicality. To this end, an extended entropy and mutual information estimator is introduced for Renyi's definitions of entropy, which encompass those of Shannon's as special cases. The mathematical properties of entropy and the proposed nonparametric estimator are investigated in light of the needs of a learning algorithm. Various Renyi's-entropy-based adaptation criteria are proposed for supervised and unsupervised training schemes and these are tested in simulations to show superiority over alternatives. Stochastic and recursive entropy estimates are introduced for on-line information theoretic data processing purposes. Simulation results obtained in various adaptation problems proved the effectiveness of these algorithms for on-line information processing. In addition, an extension to Fano's bound is proposed using Renyi's entropy definition. The extension resulted in an upper bound for the classification error probability and a lower bound. In terms of theoretical insights, these bounds also verified the intuition that in designing optimal classifiers, one needs to maximize the amount of information transferred through the classifier. The significance of this work can be summarized in one sentence as follows: The provided mathematical and algorithmic tools enable adaptive system designers to make full use of the principle of *information theoretic learning* independent of the topology of the adaptive system or the nature of the data.

CHAPTER 1
INTRODUCTION

1.1 Historical Background

The roots of adaptive systems and filtering lie deep in the early 19th century developments by Gauss and his contemporaries on function approximation theory. Although these initial attempts to understand the nature of functions (and perhaps to use this knowledge for practical problems of the time) provided a profound basis and a firm mathematical theory behind the subject, real practical advantages did not become evident until the concept of adaptive filters was established around the mid-20th century. Wiener's insightful analysis on adaptive function approximators [Wie49] and especially its application to the FIR filter structure to yield what is now known as the Wiener-Hopf equations [Hay84] provided a solid understanding of the nature of adaptive filters investigated under second-order statistical optimization criteria. The first exciting results regarding the applicability of these theoretical results came with the advent of digital computers in the 1950s and Widrow's contemporary work on the basic linear neural network structure called the adaptive linear neuron (ADALINE), which led to the well-known LMS algorithm [Wid85]. The LMS algorithm and its derivatives, for the first time, successfully allowed adaptive filters to be applied to engineering problems in real-time scenarios. As a consequence, the area of adaptive filter research gained wide acceptance by the engineering community and flourished in the following years with the contributions of numerous researchers. Eventually adaptive systems acquired the ultimate recognition of the scientific community as valuable engineering tools [Cla98, Goo84, Far98, Hay84, Nar89].

These early works, however, were mostly concentrated around the investigation of mean-square-error (MSE) and other second-order statistics as optimality criteria, and for valid reasons. Since early research mainly concentrated on linear adaptive systems, the adoption of second-order optimality measures resulted in quadratic performance surfaces, for which the analytical expression of the optimal solution could easily be obtained, allowing theoretical analyses of various aspects of the adaptation problem. Also initially, the engineering community being mostly content with the performance obtained by linear systems trained using second-order criteria, this practice carried over to the training of nonlinear adaptive systems. As a consequence of this traditional understanding that second-order statistics are sufficient to determine solutions to almost all practical engineering problems (a belief that is also backed up by the central limit theorem), for the last four decades, most of adaptive systems research rambled on similar lines; whether the adaptive system under consideration was linear or nonlinear, MSE for supervised training and other second-order criteria for unsupervised adaptation was the focus of efforts [Bis95, Dia01, Edm96, Fan01, Far98, Qur85].

More recently, on the other hand, especially in the fields of signal processing and communications, new problems were encountered that cannot be resolved by the use of mere second-order statistics, but rather necessitate the use of higher order statistical properties of the random processes involved. The problems most relevant to the scope of this work include, but are not limited to, blind source separation (BSS), independent component analysis (ICA), blind deconvolution (BD) and equalization (BE), subspace projections and data dimensionality reduction, and feature extraction and ranking [Hay00a, Hay00b, Hyv01, Nik93].

While these developments are underway in adaptive systems research, information theory emerged and flourished independently in the communications area. Although the notion of *information in the outcome of a random event* was introduced earlier by Boltzman and Hartley [Gac94, Tit00], Shannon [Sha48, Sha64] was the first to define the *average information* of a random process and to establish a profound theory around it with specific applications and implications on digital communications. While Shannon never referred to his work as *information theory*, this appealing new branch of mathematics attracted the attention and interest of many researchers, becoming a major field of research in itself from theoretical and practical viewpoints [Csi81, Cov91, Fan61, Kul68, Ren76, Ren87].

Specifically important in this research are the definitions and contributions of Alfred Renyi [Ren76], who showed that Shannon's definitions of information theoretic quantities like entropy and mutual information were in fact special cases of a more general family of definitions. These generalized definitions are called Renyi's entropy and Renyi's mutual information, etc. Although Renyi's definitions encompass Shannon's definitions as a special case for the family parameter corresponding to 1, because of the widespread recognition of Shannon's work due to its significant implications in communications theory, Renyi's work was not recognized as a useful tool by researchers in engineering and other fields until recently; most interest in Renyi's entropy is in pure information theoretical research [Bas78, Cam65, Mit75]. In the 1990s, some interest developed in Renyi's entropy in different fields including pattern recognition [Sah97], and cryptology [Cac97]. In adaptive filtering, Principe and his co-workers at CNEL made the initial attempts to break the legacy of Shannon's entropy [Fis97, Pri00a, Pri00b, Xu98, Xu99]. They successfully applied Renyi's entropy and other derivative optimality criteria to problems of blind source separation, dimensionality reduction, feature extraction, etc. Although many others used Shannon's definitions of information theoretic criteria for adaptation processes [Bel95, Ber99, Vio95], Principe was the first to introduce the terminology *information theoretic learning* (ITL) into adaptive systems literature. In this study, we extend their work, establishing a solid theory behind information theoretic learning and demonstrating superior performance in many practical applications that concern adaptive signal processing.

1.2 Brief Overview of State-of-the-Art Preceding the Current Research

The CNEL has been working on information theoretic learning for over 5 years now. Fisher [Fis97] was the first to investigate blind source separation and subspace projections from an information theoretic view point using Renyi's entropy, together with Xu [Xu99]. Although Fisher did not actually use information theoretic quantities for adaptation, his cost functions for these problems were derived as a consequence of information theoretic analyses. The use of Parzen windowing also appeared in Fisher's dissertation for the first time [Fis97], although the main focus of his work was not information theoretic learning.

Fisher was mainly interested in subspace projections that preserved most of the mutual information between the input vector and a desired reference signal. His approach should not be understood as supervised learning, because the aim of this methodology was not to match the output of the adaptive subspace projector as close as possible to the reference signal. The reference signal is merely used as a guide to extract the relevant information from the input in the context that interests the designer. The architecture used is shown in Figure 1-1. This architecture was successfully applied to SAR imagery by Fisher to extract the two most informative components from a 64x64 SAR image, where the desired output was the orientation of the vehicle in the image [Fis97].

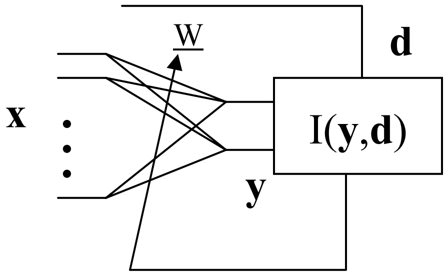


Figure 1-1. Topology used in Fisher's maximally informative subspace projections

As for the evaluation of mutual information, quadratic approximations were proposed. This is mainly because a tool to estimate the actual mutual information expression was not available to Fisher at that time. These quadratic approximations to mutual information were named QMI-ED (Quadratic Mutual Information – Euclidean Distance) and QMI-CS (Quadratic Mutual Information – Cauchy-Schwartz). The evaluation of these quantities was made possible by the use of Parzen windowing with Gaussian kernels in estimating all necessary probability density functions (pdf). Because these measures are designed such that only the integration of the squares of pdfs are required, it becomes possible to analytically express the solution to these integrals through the exploitation of the Gaussian kernels in Parzen windowing [Pri00a].

Current use of the integral of the square of a pdf necessitated the assignment of a name to this quantity; inspired by the behavior of the nonparametric estimator of this quantity based on Parzen windowing and in analogy with the potential fields generated by particles in physics they called it the *information potential*. The link with information theory is seen through the investigation of Renyi's quadratic (Order 2) entropy of a random variable X with pdf $f_X(.)$; notice the argument of the logarithm in Eq. (1.1), the definition of Renyi's quadratic entropy.

$$H_2(X) = -\log \int_{-\infty}^{\infty} f_X^2(x)dx \tag{1.1}$$

Motivated by the well-known Bell-Sejnowski algorithm for BSS [Bel95], Fisher, Wu, and Xu also applied these ideas to this problem [Fis97, Wu99, Xu99]. Consider the BSS topology shown in Figure 1-2, where **H** is the unknown mixing matrix and **W** is the separation matrix to be adapted for maximum joint output entropy. It is known that, in this square system, if the nonlinearities are matched to the cumulative density functions (cdf) of the sources, then maximizing the joint entropy at the output of the nonlinearrities will guarantee that the signals before the nonlinearities are the independent sources.

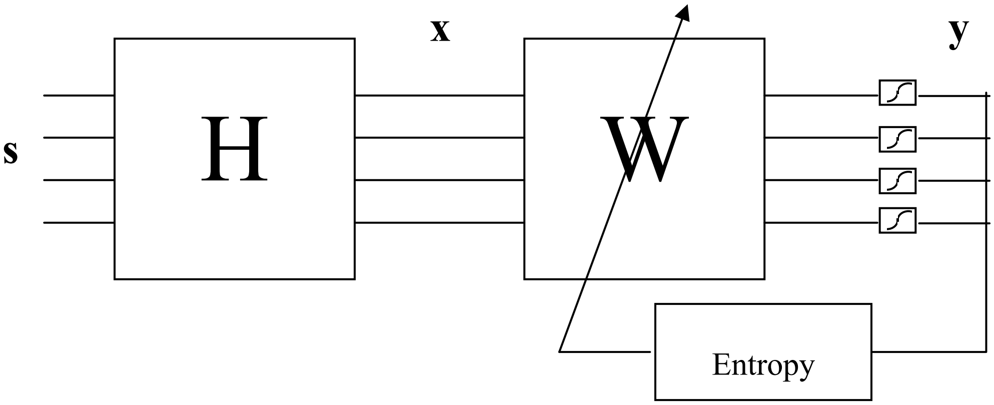


Figure 1-2. BSS topology used by Fisher's and Xu's algorithms

Noticing that the joint entropy is maximized when the joint pdf under consideration becomes uniform over its support, Fisher proposed the following criterion to be minimized.

$$J = \sum_i \left(f_Y(y_i) - 1\right)^2 \tag{1.2}$$

In this, the sample index is i and $f_Y(.)$ is the joint pdf of the output after the nonlinearities estimated by Parzen windowing. This cost function is basically the Euclidean distance between the estimated pdf and the target uniform density with the integral approximated by a summation over the samples [Fis97].

Alternatively, Xu uses Renyi's quadratic entropy directly. Noticing that in Eq. (1.1), maximizing the entropy is equivalent to minimizing the information potential, since the logarithm is a monotonic function, Xu estimates the information potential in the usual manner and adapts the separation matrix accordingly.

Both Fisher and Xu used adaptive kernel sizes to achieve global optimum and avoid local optima; they decreased the kernel size (the variance of the Gaussian kernel used in Parzen windowing) to a nominal value during the course of training starting from a large value [verbal communication with Principe]. They could not, however, determine a mathematical reason for this behavior and provided heuristic explanations as to why this phenomenon occurred.

Another application of quadratic entropy, investigated by Fisher [Fis97], is the nonlinear principal component analysis (PCA), where he maximized the output entropy of a multi-layer perceptron (MLP) to show that this procedure results in the first nonlinear principal component of a double-Gaussian distributed two-dimensional data [Pri00a].

1.3 Contributions of this Thesis to Information Theoretic Learning

Before this research had started, a nonparametric estimator for Renyi's quadratic entropy was known and used in applications by CNEL members. This nonparametric estimator, as mentioned before, was based on a Parzen window estimate of the pdf using

Gaussian kernel functions. My first contribution was to extend this entropy estimator to account for any order of Renyi's entropy and to allow the designer to use alternative suitable kernel functions, perhaps to improve performance over the Gaussian kernel counterpart [Erd02d]. Using the same idea in the derivation of the estimator for Renyi's entropy, it then becomes possible to estimate Renyi's mutual information and Renyi's divergence nonparametrically from the samples. It has been shown that the previously used quadratic entropy estimator is a special case of this generalized estimator corresponding to the entropy order choice of two and kernel function choice of Gaussian. As an immediate consequence, the idea of *information potential* is generalized to *order-α information potential* (depending on the choice of entropy order), leaving the previously defined information potential to be the quadratic information potential. Furthermore, we investigated the possibility of regarding adaptation from a particle-interaction view point. This analysis gave us a generalized family of potential-energy-functions that could be used in general for supervised or unsupervised learning. In fact, we show that some commonly used adaptation criteria fall into this class of cost functions allowing the particle interaction model to be valid for adaptation under these principles.

My second contribution was to extend the well-known Fano's bound for classifier error probability in terms of the conditional entropy of the output classes given the input classes. This classical inequality uses Shannon's definition of entropy to arrive at a lower bound for the probability of misclassification. Using Renyi's definition and applying Jensen's inequality, we obtained a family of upper bounds for the probability of error and a family of lower bounds, in which Fano's original bound resided as a special case [Erd01a, Erd02e, Erd02f]. This theoretical development complements the theory of maximally informative subspace projections for feature selection, introduced by Fisher, by showing that both lower and upper bounds for the probability of classification error depends on the amount of information transferred through the classifier.

As a third contribution, this research introduces the *minimum error entropy* (MEE) criterion for supervised adaptation. Traditionally, MSE has been the workhorse of supervised training; this has strong validating reasons behind it, including the fact that the error power is a sufficient statistic to train linear systems under Gaussian signal assumptions. It is not sufficient, however, if the objective is to extract all the relevant information from the given training data. In such situations, it is necessary to consider the amount of information lost to the error signal and it is logical to minimize this information [Erd00a, Erd02c]. This is achieved when the error entropy is minimized, for entropy is the average information of a random variable. We investigated the performance of MEE criterion in supervised training with many examples and case studies [Erd02c, Erd02d]. Comparisons with MSE showed improved performance in extracting more information from the given data. In addition, we studied the structure of the corresponding performance surface for linear adaptive systems, convergence properties in that case, and the general noise rejection capabilities of MEE [Erd01b].

The fourth contribution of this research is the *stochastic information gradient* (SIG), which allows on-line entropy manipulation on a sample-by-sample basis for real-time information theoretic adaptation of learning systems. The SIG provides a stochastic estimate for the gradient of the actual entropy of the signal under consideration and the idea behind it is identical to that of Widrow's stochastic gradient for MSE, which is known as the LMS algorithm. Also, SIG greatly simplifies the computational load required to solve for the optimal weight vector in high dimensional and large sample adaptation situations. Exploring the properties of SIG, we showed the link between a special case of SIG with Hebbian learning, and the cost functions of the form $E[\dot{e}(t)^2]$, where the dot symbolizes derivation with respect to time. Based on this last observation, for on-line adaptation scenarios, we suggested extensions to the MSE criterion involving the squares of time derivatives of the error signal (in continuous-time), in order to force the adaptive system to yield an instantaneous small error, and to continue exhibiting the same level of small error over time.

Motivated by the promising performance of SIG and the desire to improve it while maintaining the algorithmic simplicity, we examined ways of updating our entropy estimate sample-by-sample. This led to a recursive entropy estimator and this is the fifth contribution of this study. The gradient of this estimator directly yields a recursive entropy gradient, which we called the *recursive information gradient* (RIG). Remarkably, RIG loses almost nothing from the structural simplicity of SIG, yet it provides a more accurate estimate of the entropy gradient based on the acquired samples up to that point in time. Also, RIG includes SIG as a special case, corresponding to a forgetting factor of one (i.e., totally forgetting previous values).

CHAPTER 2
NONPARAMETRIC ESTIMATOR FOR RENYI'S ENTROPY

2.1 Literature Survey on Nonparametric Entropy Estimation

The problem of entropy estimation appears in many contexts in a variety of fields ranging from basic sciences like biology [Dur98] and physics [Bec93] to engineering [Cov91, Hay00a, Sha64]. From a mathematical standpoint, many approaches exist to estimate the differential entropy of a continuous random variable [Bei01]. An obvious approach, usually preferred when there is confidence that the pdf underlying the samples belongs to a known parametric family of pdfs, is to use the samples to estimate the parameters of the specific member of this family, perhaps using maximum likelihood methods, and then to evaluate the entropy of the specific pdf obtained as a result of this procedure. A useful list of explicit Shannon's entropy expressions for many commonly encountered univariate pdfs was compiled by Verdugo Lazo and Rathie [Ver78]. A similar list for various multivariate pdfs is presented by Ahmed and Gokhale [Ahm89]. This approach, although useful in entropy evaluation tasks and effective when the assumed parametric family is accurate, is not competent in adaptation scenarios, where the constantly changing pdf of the data under consideration may not lie in a simple parametric family. Then, it becomes necessary to nonparametrically estimate the entropy.

2.1.1 Plug-in Estimates

The plug-in entropy estimates are obtained by simply inserting a consistent density estimator of the data in place of the actual pdf in the entropy expression. Four types of approaches could be followed when using a plug-in estimate. The first one, named *integral estimates*, evaluates exactly or approximately the infinite integral existing in the entropy definition. Renyi's quadratic entropy estimator (developed and used successfully at CNEL) belongs to this family of entropy estimators, with an exact evaluation of the integral. An approximate estimate of this type for Shannon's entropy was also proposed [Dmi73]. Joe [Joe89] also considered an approximate integral estimate of Shannon's entropy using a kernel-based pdf estimate; however, he concluded that for multivariate cases, the approximate evaluation of the integral becomes complicated. Gyorfi and van der Meulen [Gyo87] avoid this problem by substituting a histogram estimate for the pdf.

The second approach, *resubstitution estimates*, further includes the approximation of the expectation operator in the entropy definition with the sample mean. Ahmad and Lin [Ahm76] presented a kernel-based estimate for Shannon's entropy of this type and proved the mean-square consistency of this estimate. Joe [Joe89] also considered a similar resubstitution estimate of Shannon's entropy based on kernel pdf estimates, and he concluded that in order to obtain accurate estimates especially in multivariate situations, the number of samples required increased rapidly with the dimensionality of the data. Other examples of this type of entropy estimates are more closely known to the electrical engineering community [Ber00, Bos01, Com94, Vio95, Yan97]. These estimates use spectral-estimation based or polynomial expansion type pdf estimates substituted for the actual pdf in Shannon's entropy definition, except for the last one, which uses a kernel pdf estimator. In fact, a thorough search of the literature revealed that most estimators known to the electrical engineering community concentrate on resubstitution estimates of Shannon's entropy. Depending on the specific application the authors are interested in, these estimates are tailored to suit the computational requirements desired from the algorithm. Therefore, it is possible to write out an extensive list of application-oriented references with slight differences in their entropy estimators. The nonparametric estimator for Renyi's entropy that we derive in Section 2.2 is also a member of the resubstitution class and all theoretical results in these references might apply to it after some minor modifications.

The third approach is called the *splitting data estimate*, and is similar to the resubstitution estimate, except that now the sample set is divided into two parts and one is used for density estimation while the other part is used for the sample mean [Gyo87, Gyo89, Gyo90].

Finally, the fourth approach, called *cross-validation estimate*, uses a leave-one-out principle in the resubstitution estimate. The entropy estimate is obtained by averaging the leave-one-out resubstitution estimates of the data set. Ivanov and Rozhkova [Iva81] proposed such an estimator for Shannon's entropy using a kernel-based pdf estimator.

2.1.2 Estimates Based on Sample Spacing

In this approach, a density estimate is constructed based on the sample differences. Specifically in the univariate case, if the samples are ordered from the smallest to the largest, one can define the *m*-spacing between the samples as the difference between samples that are separated by *m* samples in the ordering. This pdf estimate can then be substituted in the entropy definition as in the resubstitution estimates. Surprisingly, although the *m*-spacing density estimates might not be consistent, their corresponding *m*-spacing entropy estimates might turn out to be (weakly) consistent

[Bec93, Bei85, Hal84, Tar68]. The generalization of these estimates to multivariate cases is not trivial, however.

2.1.3 Estimates Based on Nearest Neighbor Distances

For general multivariate densities, the nearest neighbor entropy estimate is defined as the sample average of the logarithms of the normalized nearest neighbor distances plus a constant, named the Euler constant [Bec93, Koz87]. Kozachenko and Leonenko [Koz87], Tsybakov and van der Meulen [Tsy94], and Bickel and Breiman [Bic83] provide different forms of consistency for these estimates under mild conditions on the underlying densities.

2.2 Entropy Estimation for Adaptation and Learning

It is evident from the literature survey that, Shannon's entropy occupied a great deal of research time and drew much of the effort on the topic. Yet, most of the ideas about estimating Shannon's entropy are also applicable to Renyi's and other definitions of this quantity. We are mainly interested in computationally simple entropy estimators that are continuous and differentiable in terms of the samples, for our main objective here is not to estimate the entropy itself accurately, but to use this estimated quantity in optimizing the parameters of an adaptive system. Therefore, we are not strictly bounded by the convergence speeds and consistency properties of these estimators, which is mainly the concern of mathematical research on the subject. We mentioned in Section 2.1 that the previous estimator used at CNEL and the new estimator that we define in this section, are special cases of the general class of entropy estimation approaches described. First, we replicated the derivation of the old estimator for Renyi's quadratic entropy, and then we derived the extended entropy estimator, which works for any order of entropy.

2.2.1 Quadratic Entropy Estimator

The previously used nonparametric estimator for Renyi's quadratic entropy used Parzen windowing with Gaussian kernels in the following manner. Recall the definition of quadratic entropy given in Eq. (1.1) for the random variable *X*. Suppose we have *N* independent and identically distributed (iid) samples $\{x_1,\ldots,x_N\}$ from this random variable. The Parzen estimate [Par67] of the pdf using an arbitrary kernel function $\kappa_\sigma(.)$ is given in Eq. (2.1). This kernel function must be a valid pdf in general and it must be continuous and differentiable for our purposes (reasons are discussed later).

$$\hat{f}_X(x) = \frac{1}{N}\sum_{i=1}^{N}\kappa_\sigma(x - x_i) \tag{2.1}$$

Assuming Gaussian kernels, $G_\sigma(.)$, with standard deviation σ and substituting this in the quadratic entropy expression [Pri00a], we get the estimator.

$$\begin{aligned}\hat{H}_2^{old}(X) &= -\log\int_{-\infty}^{\infty}\left(\frac{1}{N}\sum_{i=1}^{N}G_\sigma(x-x_i)\right)^2 dx\\ &= -\log\frac{1}{N^2}\int_{-\infty}^{\infty}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}G_\sigma(x-x_j)\cdot G_\sigma(x-x_i)\right)dx\\ &= -\log\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\int_{-\infty}^{\infty}G_\sigma(x-x_j)\cdot G_\sigma(x-x_i)dx\\ &= -\log\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}G_{\sigma\sqrt{2}}(x_j - x_i)\end{aligned} \tag{2.2}$$

The result is easily obtained by noticing that the integral of the product of two Gaussians is exactly given by a Gaussian function whose variance is the sum of the variances of the two original Gaussian functions. Other kernel functions, however, do not result in such convenient evaluation of the integral. Nevertheless, alternative kernels might still be used in this estimator.

2.2.2 Extended Estimator for Order-α Renyi's Entropy

In the new estimator, there are no restrictions on the choice of entropy order and kernel function. Consider the definition of Renyi's order-α entropy in Eq. (2.3), which can also be written with an expectation operator [Ren70].

$$H_\alpha(X) \overset{\Delta}{=} \frac{1}{1-\alpha}\log\int_{-\infty}^{\infty} f_X^\alpha(x)dx = \frac{1}{1-\alpha}\log E_X\left[f_X^{\alpha-1}(X)\right] \tag{2.3}$$

Approximating the expectation operator with the sample mean, we get

$$H_\alpha(X) \approx \frac{1}{1-\alpha}\log\frac{1}{N}\sum_{j=1}^{N} f_X^{\alpha-1}(x_j) \tag{2.4}$$

Finally substituting the Parzen window estimator in Eq. (2.1) in Eq. (2.4) and rearranging terms, we obtain the nonparametric estimator for Renyi's entropy.

$$\begin{aligned}\hat{H}_\alpha(X) &= \frac{1}{1-\alpha}\log\frac{1}{N}\sum_{j=1}^{N}\left(\frac{1}{N}\sum_{i=1}^{N}\kappa_\sigma(x_j-x_i)\right)^{\alpha-1} \\ &= \frac{1}{1-\alpha}\log\frac{1}{N^\alpha}\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\kappa_\sigma(x_j-x_i)\right)^{\alpha-1}\end{aligned} \tag{2.5}$$

Notice that for the specific choices of $\alpha = 2$ and Gaussian kernels with standard deviation $\sigma\sqrt{2}$, Eq. (2.5) becomes identical to Eq. (2.2). In fact, for the quadratic entropy case, it is possible to make Eq. (2.5) identical to Eq. (2.2) by choosing

$$\kappa^{new}(x_j - x_i) = \int_{-\infty}^{\infty}\kappa^{old}(x-x_i)\cdot\kappa^{old}(x-x_j)dx \tag{2.6}$$

where $\kappa^{new}(.)$ denotes the kernel function used in Eq. (2.5) and $\kappa^{old}(.)$ denotes the kernel function used in Eq. (2.2). This result has an interesting implication for the estimation variance of quadratic entropy using these two estimators. Noticing that if the exact same kernel function is used in both Eq. (2.2) and Eq. (2.5), the latter has a larger estimation variance due to the additional approximation introduced by the sample mean estimation of the expectation operator. Remarkably, due to the relation in Eq. (2.6), it is possible to completely eliminate this additional variance by selecting a suitable different kernel in Eq. (2.5); specifically for the case where a Gaussian kernel is used in Eq. (2.2), this corresponds to merely increasing the kernel size by a factor of $\sqrt{2}$.

2.3 Properties of the Nonparametric Entropy Estimator

The nonparametric estimator in Eq. (2.5) is general purpose and can be used in situations where it is required to evaluate an entropy or where it is desired to adapt the weights of a learning system based on an entropic performance index. In the following, all kernel functions and random variable samples are assumed to be single-dimensional unless noted otherwise. The generalization of these results to multi-dimensional cases is trivial and the proofs follow similar lines.

Theorem 2.1. The entropy estimator in Eq. (2.5) is consistent if the Parzen windowing and the sample mean are consistent for the actual pdf of the iid samples.

Proof. The proof follows immediately from the consistency of the Parzen window estimate for the pdf [Par67] and the fact that as N goes to infinity the sample mean converges to the expected value (notice, for example, that the sample mean estimate is not consistent for infinite-variance pdfs).

This theorem is important because it points out the asymptotic limitations of the estimator. In adaptation and learning from finite samples, since we rarely have huge data sets, it is not critical for our purposes to have a consistent or an inconsistent estimate of the entropy, as long as the global optimum lies at the desired solution.

Property 2.1. The kernel size must be a parameter that satisfies the scaling property $\kappa_{c\sigma}(x) = \kappa_\sigma(x/c)/c$ for a positive scaling factor c [Par67].

This regulatory condition guarantees the kernel size affects the width of the kernel function linearly. In the analysis of the eigenstructure of the entropy cost function near the global optimum and in obtaining scale-invariant entropy-based cost functions, this property will become useful.

Property 2.2. The entropy estimator in Eq. (2.5) is invariant to the mean of the underlying density of the samples as is the actual entropy [Erd01b].

Proof. Consider two random variables X and $\overline{X}$ where $\overline{X} = X + m$ with m being a real constant. Consider the following on the entropy of $\overline{X}$.

$$\begin{aligned} H_\alpha(\overline{X}) &= \frac{1}{1-\alpha}\log\int f_{\overline{X}}^{\alpha}(\overline{x})d\overline{x} = \frac{1}{1-\alpha}\log\int f_X^{\alpha}(\overline{x}-m)d\overline{x} \\ &= \frac{1}{1-\alpha}\log\int f_X^{\alpha}(x)dx = H_\alpha(X) \end{aligned} \tag{2.7}$$

Let $\{x_1,\ldots,x_N\}$ be samples of X, then samples of $\overline{X}$ are $\{x_1+m,\ldots,x_N+m\}$. Therefore, the entropy estimate of $\overline{X}$ is

$$\begin{aligned} \hat{H}_\alpha(\overline{X}) &= \frac{1}{1-\alpha}\log\frac{1}{N^\alpha}\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\kappa_\sigma(\overline{x}_j-\overline{x}_i)\right)^{\alpha-1} \\ &= \frac{1}{1-\alpha}\log\frac{1}{N^\alpha}\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\kappa_\sigma(x_j+m-x_i-m)\right)^{\alpha-1} = \hat{H}_\alpha(X) \end{aligned} \tag{2.8}$$

Due to this property of the entropy and its estimator, in supervised learning entropy cannot be used to force the mean of the error signal to zero, for example. However, in general, since we are interested in the statistical properties of the signals other than their means, this is not a problem.

Property 2.3. The limit of Renyi's entropy as $\alpha \to 1$ is Shannon's entropy. The limit of the entropy estimator in Eq. (2.5) as $\alpha \to 1$ is Shannon's entropy estimated using Parzen windowing with sample mean approximation for expectation.

Proof. Notice that Renyi's entropy in Eq. (2.3) is discontinuous at $\alpha = 1$. However, when we take the limit of it as this parameter approaches to one, we get Shannon's entropy as shown in Eq. (2.9).

$$\begin{aligned}\lim_{\alpha\to 1} H_\alpha(X) &= \lim_{\alpha\to 1}\frac{1}{1-\alpha}\log\int f_X^\alpha(x)dx \\ &= \frac{\lim\limits_{\alpha\to 1}\int \log f_X(x)\cdot f_X^\alpha(x)dx \Big/ \int f_X^\alpha(x)dx}{\lim\limits_{\alpha\to 1} -1} \\ &= -\int f_X(x)\cdot\log f_X(x)dx = H_S(X)\end{aligned} \tag{2.9}$$

The derivation of this result for the estimator in Eq. (2.5) is shown in Eq. (2.10).

$$\begin{aligned}\lim_{\alpha\to 1}\hat{H}_\alpha(X) &= \lim_{\alpha\to 1}\frac{1}{1-\alpha}\log\frac{1}{N}\sum_{j=1}^{N}\left(\frac{1}{N}\sum_{i=1}^{N}\kappa_\sigma(x_j-x_i)\right)^{\alpha-1} \\ &= -\lim_{\alpha\to 1}\frac{\left(\frac{1}{N}\sum_{j=1}^{N}\left(\frac{1}{N}\sum_{i=1}^{N}\kappa_\sigma(x_j-x_i)\right)^{\alpha-1}\log\left(\frac{1}{N}\sum_{i=1}^{N}\kappa_\sigma(x_j-x_i)\right)\right)}{\left(\frac{1}{N}\sum_{j=1}^{N}\left(\frac{1}{N}\sum_{i=1}^{N}\kappa_\sigma(x_j-x_i)\right)^{\alpha-1}\right)} \\ &= \lim_{\alpha\to 1}\frac{-1}{N}\sum_{j=1}^{N}\log\left(\frac{1}{N}\sum_{i=1}^{N}\kappa_\sigma(x_j-x_i)\right) = \hat{H}_S(X)\end{aligned} \tag{2.10}$$

In terms of adaptation, this means that all the conclusions drawn in this research about Renyi's entropy, its estimator, and training algorithms based on Renyi's entropy apply to Shannon's definition as well, in the limit, if the entropy order approaches to one.

Proposition 2.1. In order to maintain consistency with the scaling property of the actual entropy, if the entropy estimate of samples $\{x_1,\ldots,x_N\}$ of a random variable X is estimated using a kernel size of σ, the entropy estimate of the samples $\{cx_1,\ldots,cx_N\}$ of a random variable cX must be estimated using a kernel size of $|c|\sigma$.

Proof. Consider the Renyi's entropy of the random variable cX, whose pdf is $f_X(x/c)/|c|$ in terms of the pdf of the random variable X and the scaling coefficient c.

$$H_\alpha(cX) = \frac{1}{1-\alpha}\log\int_{-\infty}^{\infty}\frac{1}{|c|}f_X^\alpha(\frac{x}{c})dx = H_\alpha(X) + \log|c| \tag{2.11}$$

Now consider the entropy estimate of the samples $\{cx_1,\ldots,cx_N\}$ using the kernel size $|c|\sigma$.

$$\begin{aligned}\hat{H}_\alpha(cX) &= \frac{1}{1-\alpha}\log\frac{1}{N^\alpha}\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\kappa_{|c|\sigma}(x_j - x_i)\right)^{\alpha-1}\\ &\frac{1}{1-\alpha}\log\frac{1}{N^\alpha}\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\frac{1}{|c|}\kappa_\sigma(\frac{cx_j - cx_i}{c})\right)^{\alpha-1}\\ &\frac{1}{1-\alpha}\log\frac{1}{|c|^{\alpha-1}N^\alpha}\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\kappa_\sigma(x_j - x_i)\right)^{\alpha-1}\\ &= \hat{H}_\alpha(X) + \log|c|\end{aligned} \tag{2.12}$$

This property is crucial to maintaining the desirable properties of entropy in adaptation when using the nonparametric estimator in place of it. For example, as we will see later in Chapter 6, the blind deconvolution problem requires a scale invariant cost function. The scaling of the kernel size as described above according to the norm of the weight vector will guarantee that the nonparametric estimation of the scale-invariant cost function possesses this property as well.

Proposition 2.2. When estimating the joint entropy of an n-dimensional random vector X from its samples $\{x_1,\ldots,x_N\}$, use a multi-dimensional kernel that is the product of single-dimensional kernels. This way, the estimate of the joint entropy and estimate of the marginal entropies are consistent.

Proof. Let the random variable X^o be the o^{th} component of X. Consider the use of single-dimensional kernels $\kappa^o_{\sigma_o}(.)$ for each of these components correspondingly. Also assume that the multi-dimensional kernel used to estimate the joint pdf of X is $\kappa_\Sigma(.)$. The Parzen estimate of the joint pdf is then given by

$$\hat{f}_X(x) = \frac{1}{N}\sum_{i=1}^{N}\kappa_\Sigma(x - x_i) \tag{2.13}$$

Similarly, the Parzen estimate of the marginal density of X^o is

$$\hat{f}_X(x) = \frac{1}{N}\sum_{i=1}^{N}\kappa^o_{\sigma_o}(x^o - x^o_i) \tag{2.14}$$

Without loss of generality, consider the marginal pdf of X^1 derived from the estimate of the joint pdf in Eq. (2.13).

$$\begin{aligned}\bar{f}_{X^1}(x^1) &= \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\hat{f}_X(x^1,\ldots,x^n)dx^2,\ldots,dx^n \\ &= \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\frac{1}{N}\sum_{i=1}^{N}\kappa_\Sigma(x^1 - x^1_i,\ldots,x^n - x^n_i)dx^2,\ldots,dx^n \\ &= \frac{1}{N}\sum_{i=1}^{N}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\kappa_\Sigma(x^1 - x^1_i,\ldots,x^n - x^n_i)dx^2,\ldots,dx^n\end{aligned} \tag{2.15}$$

Now, assuming that the joint kernel is the product of the marginal kernels evaluated at the appropriate values, i.e., $\kappa_\Sigma(x) = \prod_{o=1}^{n}\kappa_{\sigma_o}(x^o)$, we get Eq. (2.16). Thus, this choice of the multi-dimensional kernel for joint entropy estimation guarantees consistency between the joint and marginal pdf and entropy estimates. This property is, in fact, critical for the general pdf estimation problem besides being important in entropy estimation.

$$\begin{aligned}\bar{f}_{X^1}(x^1) &= \frac{1}{N}\sum_{i=1}^{N}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\prod_{o=1}^{n}\kappa_{\sigma_o}^{o}(x^o - x_i^o)dx^2,\ldots,dx^n \\ &= \frac{1}{N}\sum_{i=1}^{N}\kappa_{\sigma_1}^{1}(x^1 - x_i^1)\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\prod_{o=2}^{n}\kappa_{\sigma_o}^{o}(x^o - x_i^o)dx^2,\ldots,dx^n \\ &= \frac{1}{N}\sum_{i=1}^{N}\kappa_{\sigma_1}^{1}(x^1 - x_i^1)\left(\prod_{o=2}^{n}\int_{-\infty}^{\infty}\kappa_{\sigma_o}^{o}(x^o - x_i^o)dx^o\right) \\ &= \frac{1}{N}\sum_{i=1}^{N}\kappa_{\sigma_1}^{1}(x^1 - x_i^1) = \hat{f}_{X^1}(x^1)\end{aligned} \tag{2.16}$$

This important issue should be considered in adaptation scenarios where the marginal entropies of multiple signals and their joint entropy are used in the cost function simultaneously. It is desirable to have consistency between the marginal and joint entropy estimates.

Theorem 2.2. If the maximum value of the kernel $\kappa_\sigma(\xi)$ is achieved when $\xi = 0$, then the minimum value of the entropy estimator in Eq. (2.5) is achieved when all samples are equal to each other, i.e., $x_1 = \ldots = x_N = c$ [Erd02d].

Proof. By substitution, we find that the entropy estimator takes the value $-\log\kappa_\sigma(0)$ when all samples are equal to each other. We need to show that

$$\frac{1}{1-\alpha}\log\frac{1}{N^\alpha}\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\kappa_\sigma(x_j - x_i)\right)^{\alpha-1} \geq -\log\kappa_\sigma(0) \tag{2.17}$$

For $\alpha > 1$, this is equivalent to showing that

$$\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\kappa_\sigma(x_j - x_i)\right)^{\alpha-1} \leq N^\alpha\kappa_\sigma^{\alpha-1}(0) \tag{2.18}$$

Replacing the left hand side of Eq. (2.18) with its upper bound we get Eq. (2.19). Since the kernel function is chosen such that its maximum occurs when its argument is zero, we obtain the desired result given in Eq. (2.18).

$$
\begin{aligned}
&\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\kappa_\sigma(x_j-x_i)\right)^{\alpha-1} \le N\max_j\left[\left(\sum_{i=1}^{N}\kappa_\sigma(x_j-x_i)\right)^{\alpha-1}\right] \\
&\le N\max_j\left[N^{\alpha-1}\max_i \kappa_\sigma^{\alpha-1}(x_j-x_i)\right]_{i} = N^{\alpha}\max_{i,j}\kappa_\sigma^{\alpha-1}(x_j-x_i)
\end{aligned}
\tag{2.19}
$$

The proof for the case $\alpha < 1$ is similar. It uses the min operator instead of max due to the direction of the inequality.

In supervised training, it is imperative that the cost function achieves its global minimum when all the error samples are zero. Minimum error entropy learning using this entropy estimator, which will be introduced in Chapter 4, will become a valid supervised training approach with this property of the entropy estimator. In addition, the unsupervised training scenarios like minimum entropy blind deconvolution, which will be discussed in Chapter 6, will benefit from this property of the estimator.

Theorem 2.3. If the kernel function $\kappa_\sigma(.)$ is continuous, differentiable, symmetric and unimodal, then the global minimum described in Theorem 2.2 of the entropy estimator in Eq. (2.5) is smooth, i.e., it has a zero gradient and a positive semi-definite Hessian matrix. The Hessian is semi-definite because there is one eigenvector corresponding to the direction that only changes the mean of data, along which we know the entropy estimator to be constant due to *Prop. 2.2.* Notice that under the listed conditions, the maximum value of the kernel is achieved when its argument is zero.

Proof. Let $\bar{x} = [x_1 \quad \ldots \quad x_N]^T$ be the data samples collected in a vector for notational simplicity. Without loss of generality, consider the data set given by $\bar{x} = 0$, meaning all samples are zero. With some algebra, the gradient and the Hessian matrix of the expression in Eq. (2.5) with respect to $\bar{x}$ are found as

$$\begin{aligned}\frac{\partial \hat{H}_\alpha}{\partial x_k} &= \frac{1}{1-\alpha}\frac{\partial \hat{V}_\alpha / \partial x_k}{\hat{V}_\alpha} \\ \frac{\partial^2 \hat{H}_\alpha}{\partial x_l \partial x_k} &= \frac{1}{1-\alpha}\frac{(\partial^2 \hat{V}_\alpha / \partial x_l \partial x_k)\hat{V}_\alpha - (\partial \hat{V}_\alpha / \partial x_k)(\partial \hat{V}_\alpha / \partial x_l)}{\hat{V}_\alpha^2}\end{aligned} \tag{2.20}$$

where the variable $\hat{V}_\alpha$ is the argument of the logarithm in the final expression in Eq. (2.5). Evaluating these expressions at $\bar{x} = 0$, we get

$$\begin{aligned}\hat{V}_\alpha\Big|_{\bar{x}=0} &= \kappa_\sigma^{\alpha-1}(0) \\ \frac{\partial \hat{V}_\alpha}{\partial x_k}\Bigg|_{\bar{x}=0} &= \frac{(\alpha-1)}{N^\alpha}\left[N^{\alpha-1}\kappa_\sigma^{\alpha-2}(0)\kappa'(0) - N^{\alpha-1}\kappa_\sigma^{\alpha-2}(0)\kappa'(0)\right] = 0 \\ \frac{\partial^2 \hat{V}_\alpha}{\partial x_k^2}\Bigg|_{\bar{x}=0} &= \frac{(\alpha-1)(N-1)\kappa_\sigma^{\alpha-3}(0)}{N^2}\left[(\alpha-2)\kappa'^2(0) + 2\kappa(0)\kappa''(0)\right] \\ \frac{\partial^2 \hat{V}_\alpha}{\partial x_l\,\partial x_k}\Bigg|_{\bar{x}=0} &= -\frac{(\alpha-1)\kappa_\sigma^{\alpha-3}(0)}{N^2}\left[(\alpha-2)\kappa'^2(0) + 2\kappa(0)\kappa''(0)\right]\end{aligned} \tag{2.21}$$

which shows that the gradient vector is zero and that the Hessian matrix is composed of

$$\frac{\partial^2 \hat{H}_\alpha}{\partial x_l \partial x_k}\Bigg|_{\bar{x}=0} = \begin{cases} -(N-1)\kappa_\sigma^{-\alpha-1}(0)\left[(\alpha-2)\kappa'^2(0) + 2\kappa(0)\kappa''(0)\right]/N^2, l = k \\ \kappa_\sigma^{-\alpha-1}(0)\left[(\alpha-2)\kappa'^2(0) + 2\kappa(0)\kappa''(0)\right]/N^2, l \neq k \end{cases} \tag{2.22}$$

Denoting the diagonal terms by *a* and the off diagonal terms by *b*, we can determine all the eigenvalue-eigenvector pairs of this matrix to be

$$\{0,[1,...,1]^T\}, \{aN/(N-1),[1,-1,0,...,0]^T\}, \{aN/(N-1),[1,0,-1,0,...,0]^T\},... \tag{2.23}$$

Notice that the non-zero eigenvalue has a multiplicity of *N*-1 and for a kernel function as described in the theorem and for *N*>1 this eigenvalue is positive, since the kernel evaluated at zero is positive, the first derivative of the kernel evaluated at zero is zero and the second derivative is negative. Thus the Hessian matrix at the global minimum of the entropy estimator is positive semi-definite.

In adaptation using numerical optimization techniques, it is crucial that the global optimum is a smooth point in the weight space with zero gradient and finite-eigenvalue Hessian. This last theorem shows that the nonparametric estimator is suitable for entropy minimization adaptation scenarios using such approaches.

Property 2.4. If the kernel function satisfies the conditions in Theorem 2.3, then in the limit, as the kernel size tends to infinity, the quadratic entropy estimator approaches to the logarithm of a scaled and biased version of the sample variance.

Proof. Let $\{x_1,\ldots,x_N\}$ be the samples of X. We denote the second-order sample moment and the sample mean with the following.

$$\begin{aligned} \overline{x^2} &= \frac{1}{N}\sum_{j=1}^{N} x_j^2 \\ \bar{x}^2 &= \left(\frac{1}{N}\sum_{j=1}^{N} x_j\right)^2 \end{aligned} \tag{2.24}$$

Since by assumption the kernel size is very large, the pair-wise differences of samples will be very small compared to the kernel size, thus allowing the second-order Taylor series expansion of the kernel function around zero to be a valid approximation. Also, due to the kernel function being symmetric and differentiable, its first order derivative at zero will be zero yielding

$$\kappa_\sigma(\xi) \approx \kappa_\sigma(0) + \kappa'_\sigma(0)\xi + \kappa''_\sigma(0)\xi^2/2 = \kappa_\sigma(0) + \kappa''_\sigma(0)\xi^2/2 \tag{2.25}$$

Substituting this in the quadratic entropy estimator obtained from Eq. (2.5) by substituting $\alpha = 2$, we get Eq. (2.26), where $\overline{x^2} - \bar{x}^2$ is the sample variance. Notice that the kernel size affects the scale factor multiplying the sample variance in Eq. (2.26). In addition to this, there is a bias depending on the kernel's center value.

$$
\begin{aligned}
\hat{H}_2(X) &\approx -\log\left[\frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}\left(\kappa_\sigma(0)+\kappa''_\sigma(0)(x_j-x_i)^2/2\right)\right]\\
&= -\log\left[\kappa_\sigma(0)+\frac{1}{2}\kappa''_\sigma(0)\left(\frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}\left(x_j^2-2x_jx_i+x_i^2\right)\right)\right] \qquad (2.26)\\
&-\log\left[\kappa_\sigma(0)+\frac{1}{2}\kappa''_\sigma(0)\left(\overline{x^2}-\overline{x}^2\right)\right]
\end{aligned}
$$

Property 2.5. In the case of joint entropy estimation, if the multi-dimensional kernel function satisfies $\kappa_\Sigma(\xi)=\kappa_\Sigma(R^{-1}\xi)$ for all orthonormal matrices, *R*, then the entropy estimator in Eq. (2.5) is invariant under rotations as is the actual entropy of a random vector *X*. Notice that the condition on the joint kernel function requires hyper-spherical symmetry.

Proof. Consider two *n*-dimensional random vectors *X* and $\overline{X}$ related to each other with $\overline{X}=RX$ where *R* is an *n*x*n* real orthonormal matrix. Then the entropy of $\overline{X}$ is

$$
\begin{aligned}
H_\alpha(\overline{X}) &= \frac{1}{1-\alpha}\log\int_{-\infty}^{\infty} f_{\overline{X}}^{\alpha}(\overline{x})d\overline{x} = \frac{1}{1-\alpha}\log\int_{-\infty}^{\infty}\frac{1}{|R|^{\alpha}}f_X^{\alpha}(R^{-1}\overline{x})d\overline{x}\\
&= \frac{1}{1-\alpha}\log\int_{-\infty}^{\infty}\frac{1}{|R|^{\alpha}}f_X^{\alpha}(x)|R|dx = \frac{1}{1-\alpha}\log|R|^{1-\alpha}\int_{-\infty}^{\infty}f_X^{\alpha}(x)dx \qquad (2.27)\\
&= H_\alpha(X)+\log|R| = H_\alpha(X)
\end{aligned}
$$

Now consider the estimation of the joint entropy of $\overline{X}$ from its samples, which are given by $\{Rx_1,\ldots,Rx_N\}$, where $\{x_1,\ldots,x_N\}$ are samples of *X*. Suppose we use a multi-dimensional kernel $\kappa_\Sigma(.)$ that satisfies the required condition. This results in the derivation in Eq. (2.28). In adaptation scenarios where the invariance-under-rotations property of entropy needs to be exploited, the careful choice of the joint kernel becomes important. Property 2.5 describes how to select kernel functions in such situations.

$$\begin{aligned}\hat{H}_\alpha(\bar{X}) &= \frac{1}{1-\alpha}\log\frac{1}{N^\alpha}\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\kappa_\Sigma(Rx_j - Rx_i)\right)^{\alpha-1}\\ &= \frac{1}{1-\alpha}\log\frac{1}{N^\alpha}\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\frac{1}{|R|}\kappa_\Sigma(R^{-1}(Rx_j - Rx_i))\right)^{\alpha-1}\\ &= \frac{1}{1-\alpha}\log|R|^{\alpha-1}\frac{1}{N^\alpha}\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\kappa_\Sigma(x_j - x_i)\right)^{\alpha-1}\\ &= \hat{H}_\alpha(X)\end{aligned} \tag{2.28}$$

Theorem 2.4. $\lim_{N\to\infty}\hat{H}_\alpha(X) = H_\alpha(\hat{X}) \geq H_\alpha(X)$, where $\hat{X}$ is a random variable with the pdf $f_X(.) * \kappa_\sigma(.)$. The equality (in the inequality portion) occurs if and only if (iff) the kernel size is zero. This result is also valid on the average for the finite-sample case.

Proof. It is well known that the Parzen window estimate of the pdf of X converges consistently to $f_X(.) * \kappa_\sigma(.)$. Therefore, the entropy estimator in Eq. (2.5) converges to the actual entropy of this pdf. To prove the inequality consider

$$e^{(1-\alpha)H_\alpha(\hat{X})} = \int_{-\infty}^{\infty} p_{\hat{X}}^{\alpha}(y)dy = \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty}\kappa_\sigma(\tau)f_X(y-\tau)d\tau\right]dy \tag{2.29}$$

Using Jensen's inequality for convex and concave cases, we get Eq. (2.30), where we defined the mean-invariant quantity $V_\alpha(X)$ as the integral of the α^{th} power of the pdf of X, which is the argument of the log in the definition of Renyi's entropy given in Eq. (2.3). Reorganizing the terms in Eq. (2.30) and using the relationship between entropy and information potential, regardless of the value of α and the direction of the inequality, we arrive at the conclusion $H_\alpha(\hat{X}) \geq H_\alpha(X)$. The fact that these results are also valid on the average for the finite-sample case is due to the property $E[\hat{f}_X(.)] = f_X(.) * \kappa_\sigma(.)$

of Parzen windowing, which relates the average pdf estimate to the actual value and the kernel function.

$$\begin{aligned}\exp((1-\alpha)H_\alpha(\hat{X})) &\overset{\alpha>1}{\le}\left(\overset{\alpha<1}{\ge}\right)\int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty}\kappa_\sigma(\tau)[f_X(y-\tau)]^\alpha\,d\tau\right]dy \\ &= \int_{-\infty}^{\infty}\kappa_\sigma(\tau)\left[\int_{-\infty}^{\infty}[f_X(y-\tau)]^\alpha\,dy\right]d\tau = \int_{-\infty}^{\infty}\kappa_\sigma(\tau)V_\alpha(X)d\tau \\ &= V_\alpha(X)\cdot\int_{-\infty}^{\infty}\kappa_\sigma(\tau)d\tau = V_\alpha(X)\end{aligned} \tag{2.30}$$

This theorem will be useful in proving asymptotic noise rejection properties of the entropy-based adaptation criteria, and showing that for entropy minimization, the proposed estimator provides a useful approximation in the form of an upper bound to the true entropy of the signal under consideration.

2.4 Renyi's Divergence Measure

Closely related to Shannon's entropy, Kullback-Leibler divergence is a commonly used information theoretic distance measure to measure the divergence between two pdfs [Kul68]. For two arbitrary pdfs *q*(*x*) and *p*(*x*), the K-L divergence of *q*(*x*) from *p*(*x*) is defined as

$$D_{KL}(q;p)\overset{\Delta}{=}\int_{-\infty}^{\infty}q(x)\log\frac{q(x)}{p(x)}dx \tag{2.31}$$

Notice that the K-L divergence is not symmetric with respect to its arguments and its minimum value of zero is attained iff the two pdfs are identically equal to each other. This divergence measure finds use in Shannon's mutual information definition and applications in many engineering problems that require testing the hypothesis of the equality/inequality of two pdfs.

It is possible to extend the K-L divergence to a family of distance measures, which we will call *Renyi's divergence*. The definition of this divergence measure and some of its basic properties are given in

Theorem 2.4. Renyi's order-α divergence of $q(x)$ from $p(x)$ is defined as

$$D_\alpha(q;p) \overset{\Delta}{=} \frac{1}{\alpha-1}\log\int_{-\infty}^{\infty} q(x)\left(\frac{q(x)}{p(x)}\right)^{\alpha-1} dx \tag{2.32}$$

and some of its properties are as follows:

i. $D_\alpha(q;p) \geq 0, \quad \forall p, q, \alpha > 0$

ii. $D_\alpha(q;p) = 0$ iff $p(x) = q(x) \ \forall x \in \Re$

iii. $\lim_{\alpha\to 1} D_\alpha(q;p) = D_{KL}(q;p)$

Proof. We do the proof of each part separately.

i. Using Jensen's inequality on the argument of the logarithm in Eq. (2.32), we get

$$\int_{-\infty}^{\infty} q(x)\left(\frac{p(x)}{q(x)}\right)^{1-\alpha} dx \ \underset{0<\alpha<1}{\overset{\alpha>1}{\gtrless}} \ \left(\int_{-\infty}^{\infty} q(x)\left(\frac{p(x)}{q(x)}\right)dx\right)^{1-\alpha} = 1 \tag{2.33}$$

Substituting this result in Eq. (2.32), we obtain the desired inequality for all values of $\alpha>$ 0.

ii. Clearly, if $q(x) = p(x)$, then $D_\alpha(q;p) = 0$. For the reverse direction, suppose we are given that $D_\alpha(q;p) = 0$. Assume $q(x) \neq p(x)$, so that we can write $p(x) = q(x) + \delta(x)$, where $\int_{-\infty}^{\infty} \delta(x) = 0$, and $\exists x \in \Re$ such that $\delta(x) \neq 0$. Consider the divergence between these two pdfs, which is shown in Eq. (2.34). Starting by equating this divergence to zero, we obtain

$$\begin{aligned} D_\alpha(q;p) &= \frac{1}{1-\alpha}\log\int_{-\infty}^{\infty} q(x)\left(\frac{q(x)+\delta(x)}{q(x)}\right)^{1-\alpha} dx \\ &= \frac{1}{1-\alpha}\log\int_{-\infty}^{\infty} q(x)\left(1+\frac{\delta(x)}{q(x)}\right)^{1-\alpha} dx = 0 \end{aligned} \tag{2.34}$$

which implies that

$$\int_{-\infty}^{\infty} q(x)\left(1+\frac{\delta(x)}{q(x)}\right)^{1-\alpha} dx = 1 \ \Rightarrow\ \left(1+\frac{\delta(x)}{q(x)}\right) = 1,\ \ \forall x \in \Re \tag{2.35}$$

From this last result, we conclude that $\delta(x)=0,\ \forall x \in \Re$, which contradicts our initial assumption, therefore, we conclude that $q(x)=p(x)$.

iii. Consider the limit of Eq. (2.32) as $\alpha \to 1$.

$$\begin{aligned} \lim_{\alpha\to 1} D_\alpha(q;p) &= \lim_{\alpha\to 1}\frac{1}{\alpha-1}\log\int_{-\infty}^{\infty} q(x)\left(\frac{p(x)}{q(x)}\right)^{1-\alpha} dx \\ &= \frac{\lim\limits_{\alpha\to 1}\int_{\infty-}^{\infty} -q(x)\left(\frac{p(x)}{q(x)}\right)^{1-\alpha}\log\left(\frac{p(x)}{q(x)}\right)dx}{\lim\limits_{\alpha\to 1}\int_{\infty-}^{\infty} q(x)\left(\frac{p(x)}{q(x)}\right)^{1-\alpha} dx} \\ &= \int_{\infty-}^{\infty} q(x)\log\left(\frac{q(x)}{p(x)}\right)dx = D_{KL}(q;p) \end{aligned} \tag{2.36}$$

Following the same ideas used in deriving the estimator for Renyi's entropy, we can determine a kernel-based resubstitution estimate of Renyi's order-α divergence. Suppose we have the iid samples $\{x_1^q,...,x_N^q\}$ and $\{x_1^p,...,x_M^p\}$ drawn from $q(x)$ and $p(x)$, respectively. The nonparametric estimator for Renyi's divergence obtained with this approach is given in Eq. (2.37). Notice that the computational complexity is again $O(N^2)$, the same as the entropy estimator. The ratio of sums, however, complicates the gradient expression.

$$
\begin{aligned}
D_\alpha(q;p) &= \frac{1}{\alpha-1}\log E_q\left[\left(\frac{q(x)}{p(x)}\right)^{\alpha-1}\right] \approx \frac{1}{\alpha-1}\log\frac{1}{N}\sum_{j=1}^{N}\left(\frac{\hat{q}(x_j^q)}{\hat{p}(x_j^q)}\right)^{\alpha-1} \\
&\frac{1}{\alpha-1}\log\frac{1}{N}\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N}\kappa^q(x_j^q-x_i^q)}{\sum_{i=1}^{m}\kappa^p(x_j^q-x_i^p)}\right)^{\alpha-1} = \hat{D}_\alpha(q;p)
\end{aligned}
\tag{2.37}
$$

Recall that Shannon's mutual information between the components of an n-dimensional random vector X is equal to the K-L divergence of the joint distribution of X from the product of the marginal distributions of the components of X [Cov91]. Similarly, Renyi's order-α mutual information is defined as the Renyi's divergence between the same quantities. Letting $f_X(.)$ be the joint distribution and $f_{X^o}(.)$ to be the marginal density of the oth component, Renyi's mutual information becomes [Ren76]

$$
I(X) \overset{\Delta}{=} \frac{1}{\alpha-1}\log\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\frac{f_X^\alpha(x^1,...,x^n)}{\prod_{o=1}^{n} f_{X^o}^{\alpha-1}(x^o)}dx^1...dx^n \tag{2.38}
$$

Once again, it is possible to write a kernel-based resubstitution estimator for Renyi's mutual information by approximating the joint expectation with a sample mean

$$
\begin{aligned}
I_\alpha(X) &\overset{\Delta}{=} \frac{1}{\alpha-1}\log E_X\left[\left(\frac{f_X(x^1,...,x^n)}{\prod_{o=1}^{n} f_{X^o}(x^o)}\right)^{\alpha-1}\right] \\
&\approx \frac{1}{\alpha-1}\log\frac{1}{N}\sum_{j=1}^{N}\left(\frac{f_X(x_j)}{\prod_{o=1}^{n} f_{X^o}(x_j^o)}\right)^{\alpha-1}
\end{aligned}
\tag{2.39}
$$

and then replacing the pdfs with their Parzen estimators that use consistent kernels between the marginal and joint pdf estimates as mentioned in Proposition 2.2, we get the nonparametric mutual information estimator in Eq. (2.40).

$$\begin{aligned}\hat{I}_\alpha(X) &\overset{\Delta}{=} \frac{1}{\alpha-1}\log\frac{1}{N}\sum_{j=1}^{N}\left(\frac{\left(\frac{1}{N}\sum_{i=1}^{N}\kappa_\Sigma(x_j-x_i)\right)}{\prod_{o=1}^{n}\left(\frac{1}{N}\sum_{i=1}^{N}\kappa_{\sigma_o}(x_j^o-x_i^o)\right)}\right)^{\alpha-1} \\ &= \frac{1}{\alpha-1}\log\frac{1}{N}\sum_{j=1}^{N}\left(\frac{\left(\frac{1}{N}\sum_{i=1}^{N}\prod_{o=1}^{n}\kappa_{\sigma_o}(x_j^o-x_i^o)\right)}{\prod_{o=1}^{n}\left(\frac{1}{N}\sum_{i=1}^{N}\kappa_{\sigma_o}(x_j^o-x_i^o)\right)}\right)^{\alpha-1}\end{aligned} \tag{2.40}$$

This nonparametric mutual information estimator is general purpose and it can be used in problems where it is necessary to evaluate the mutual information between sets of samples and in adaptation scenarios where optimizing according to the mutual information between certain variables is the objective. Due to Theorem 2.4, the limit of Eq. (2.40) is an estimate of Shannon's mutual information between the random variables under consideration.

Another possibly useful divergence measure between the pdfs $q(x)$ and $p(x)$ is the Csiszar divergence defined as [Csi81]

$$D_g(q;p) \overset{\Delta}{=} \int_{-\infty}^{\infty} q(x)g\left(\frac{p(x)}{q(x)}\right)dx \tag{2.41}$$

where the function $g(.)$ has to be convex with $g(1) = 0$. The K-L divergence is also a special case of this divergence measure corresponding to the choice $g(.) = -\log(.)$. A nonparametric resubstitution estimator of the form in Eq. (2.37) could also be derived using the same principles for the Csiszar divergence between two pdfs.

In general, it is possible to use all the estimators derived in this chapter, given in Eq. (2.5), Eq. (2.37), and Eq. (2.40), in evaluation or adaptation situations, where it is necessary to operate based on entropy, divergence, or mutual information.

One final note on the *trick* of replacing the expected value with the sample mean: This approach could, in general, be taken to evaluate definite integrals of the form $\int_a^b g(x)dx$ simply by rewriting it as $\int_a^b q(x)\frac{g(x)}{q(x)}dx = E_q\left[\frac{g(x)}{q(x)}\right]$, where $q(x)$ is chosen such that it is a valid pdf with support $[a,b]$ and it is easy generate random samples that obey this distribution law. In fact, the equi-interval Riemann sum approximation to this definite integral could be regarded as a special case that corresponds to the choice of uniform distribution for $q(x)$. The general name for this approach of evaluating definite integrals is *stochastic integration*.

CHAPTER 3
PARTICLE INTERACTION MODEL FOR LEARNING FROM SAMPLES

3.1 Quadratic Information Potential and Quadratic Information Forces

The idea of regarding the samples as *information particles* was introduced by Principe *et al.* [Pri00a] upon the observation that under minimization or maximization of entropy adaptation rule, the gradient of the weight vector comprised of two sub-quantities: the sensitivity of the output of the adaptive network with respect to its weights and the sensitivity of the overall value of the performance index on the values of the samples, which seemed to interact with each other through laws that resembled the potential fields and their associated forces in physics. Through this analogy, which was made possible by the kernel estimator for quadratic entropy they were using, they named the samples as information particles and their interaction through the formulation of the entropy estimator and the selected kernel function became the main focus of information theoretic learning [Pri00a].

Recall the quadratic entropy definition they used in Eq. (1.1). Since the logarithm is a monotonically increasing function, maximization/minimization of the quadratic entropy can be equivalently reduced to minimization/maximization of the argument of the log, which is the integral of the square of the pdf under consideration. Let's denote this quantity by $V_2(X)$. Remember from Eq. (2.2) that the nonparametric estimator for this quantity is given by the following double summation in Eq. (3.1). We consider this quantity as a summation of contributions from each particle x_j.

$$\hat{V}_2^{old}(X) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G_{\sigma\sqrt{2}}(x_j - x_i) \tag{3.1}$$

Let's denote this contribution for this corresponding sample with $V_2^{old}(x_j)$. Then we have

$$V_2^{old}(x_j) \overset{\Delta}{=} \frac{1}{N^2}\sum_{i=1}^{N} G_{\sigma\sqrt{2}}(x_j - x_i) \tag{3.2}$$

Noting that the derivative of the Gaussian evaluated at zero is zero, the derivative of this contribution with respect to the value of this sample is easily evaluated to be

$$\frac{\partial}{\partial x_j}\hat{V}_2^{old}(x_j) = \frac{1}{N^2}\sum_{i=1}^{N} G'_{\sigma\sqrt{2}}(x_j - x_i) \tag{3.3}$$

Now, we can regard this derivative as a contribution of derivatives due to all other samples and denoting the contribution by sample x_i with $F_2^{old}(x_j \mid x_i)$, and the overall derivative with respect to x_j with $F_2^{old}(x_j)$, we get

$$\begin{aligned} F_2^{old}(x_j \mid x_i) &\overset{\Delta}{=} \frac{1}{N^2} G'_{\sigma\sqrt{2}}(x_j - x_i) \\ F_2^{old}(x_j) &\overset{\Delta}{=} \frac{\partial}{\partial x_j} V_2^{old}(x_j) = \sum_{i=1}^{N} F_2^{old}(x_j \mid x_i) \end{aligned} \tag{3.4}$$

Principe *et al.* named these two quantities as the *information force on sample x_j due to sample x_i* and the *total information force acting on sample x_j*. Due to the differentiation relationship between $F_2^{old}(x_j)$ and $V_2^{old}(x_j)$, the latter is called the *total information potential of particle x_j* and $V_2^{old}(X)$, being the sum of all the information potentials of the individual information particles, is named as the *overall information potential of the sample set*.

An illustration of the information forces in single- and multi-dimensional situations is provided by Principe *et al.* [Pri00a] with an explanation of their roles in the adaptation of the weights of the learning system for a number of applications.

3.2 Extension to Order-α Information Potential and Information Forces

Recall the definition of Renyi's order-α entropy given in Eq. (2.3). We will simply name the argument of the logarithm as the order-α information potential. Thus, for a random variable X with pdf $f_X(.)$ the information potential is

$$V_\alpha(X) \overset{\Delta}{=} \int_{-\infty}^{\infty} f_X^\alpha(x)dx \tag{3.5}$$

Its nonparametric estimator is given in Eq. (2.5) as

$$\hat{V}_\alpha(X) = \frac{1}{N^\alpha}\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\kappa_\sigma(x_j - x_i)\right)^{\alpha-1} \tag{3.6}$$

which can be written as a sum of contributions from each sample x_j, denoted $\hat{V}_\alpha(x_j)$

$$\begin{aligned}\hat{V}_\alpha(x_j) &\overset{\Delta}{=} \frac{1}{N^\alpha}\left(\sum_{i=1}^{N}\kappa_\sigma(x_j - x_i)\right)^{\alpha-1}\\ \hat{V}_\alpha(X) &= \sum_{j=1}^{N}\hat{V}_\alpha(x_j)\end{aligned} \tag{3.7}$$

Naturally (in analogy with physical potentials), we determine the order-α information forces by simply taking the derivative of these information potentials with respect to the particle location (sample value).

$$\begin{aligned}\hat{F}_\alpha(x_j) &\overset{\Delta}{=} \frac{\partial}{\partial x_j}\hat{V}_\alpha(x_j) = \frac{\alpha-1}{N^\alpha}\left(\sum_{i=1}^{N}\kappa_\sigma(x_j - x_i)\right)^{\alpha-2}\left(\sum_{i=1}^{N}\kappa'_\sigma(x_j - x_i)\right)\\ &= (\alpha-1)\hat{f}_X^{\alpha-2}(x_j)\hat{F}_2(x_j)\end{aligned} \tag{3.8}$$

The total information force acting on particle x_j is found to be as in Eq. (3.8) (assuming that the kernel's derivative at zero is zero), where the quadratic information force is similar to Eq. (3.4), with the exception that the kernel function need not be specifically Gaussian. In Eq. (3.8), the quadratic force is defined as

$$\hat{F}_2(x_j) \overset{\Delta}{=} \frac{1}{N^2}\left(\sum_{i=1}^{N} \kappa'_\sigma(x_j - x_i)\right) \tag{3.9}$$

From Eq. (3.8), which is the total information force acting on article x_j, and using the additivity of quadratic forces in Eq. (3.9) as in Eq. (3.4), we can write out the individual contributions of every other sample as

$$\hat{F}_\alpha(x_j \mid x_i) = (\alpha - 1)\hat{f}_X^{\alpha-2}(x_j)\hat{F}_2(x_j \mid x_i) \tag{3.10}$$

where we defined

$$\hat{F}_2(x_j \mid x_i) \overset{\Delta}{=} \frac{1}{N^2}\kappa'_\sigma(x_j - x_i) \tag{3.11}$$

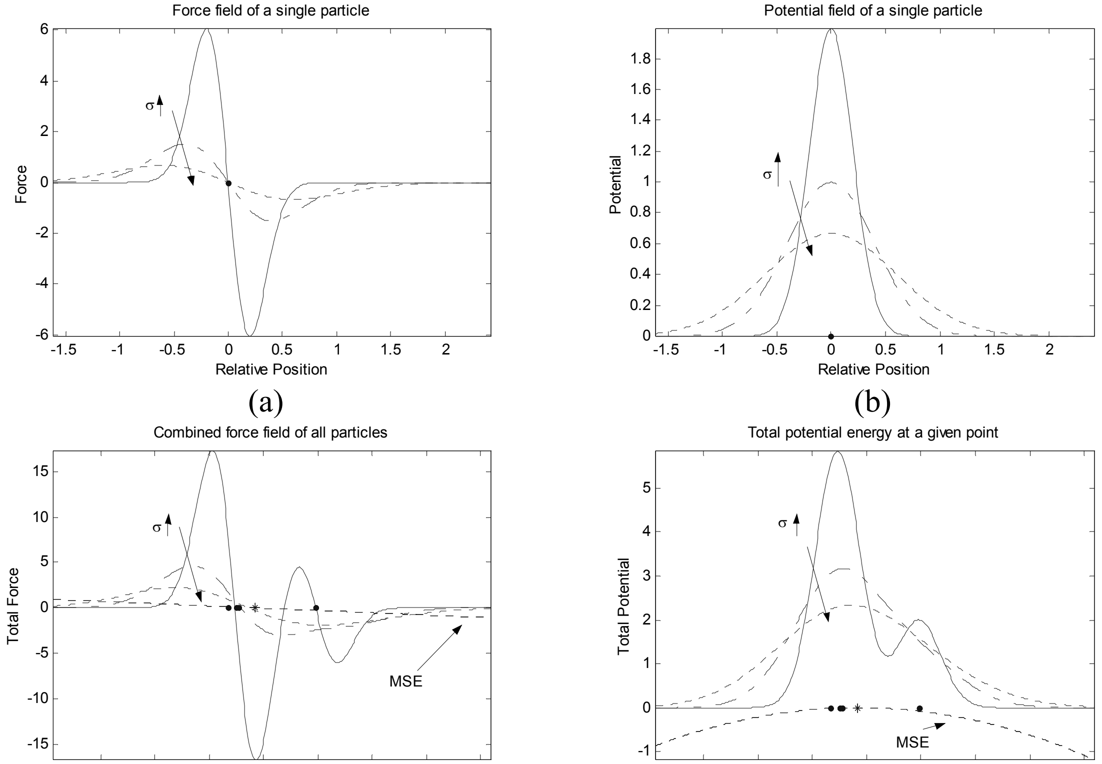
Although we considered above only the single-dimensional case, extensions of these information potential and information force definitions to multi-dimensional situations is trivial. Note that, in choosing multi-dimensional kernel functions, some restrictions apply as mentioned in Chapter 2.

Notice that the generalized information forces introduce a scaling factor that depends on the estimated probability density of the corresponding particle and the selected entropy order. Specifically, for $\alpha > 2$, the scale factor (power of the estimated pdf) in Eq. (3.8) becomes a monotonically increasing function of the pdf value, meaning that compared to the quadratic case, the forces experienced by those samples with larger probability density (in dense regions of the sample space) will be amplified. For $\alpha < 2$, on the other hand, the opposite will take place, and the forces on sparse regions of the data will be amplified.

In addition, we notice from Eq. (3.8) that the force becomes zero for the choice of α=1. This does not mean that the force is zero for Shannon's entropy choice. First of all, the information potential as we defined it is constant at 1 for all pdfs and Renyi's entropy is discontinuous at this value of the entropy order; therefore the direct substitution of this value in the expressions should be avoided in general. However, a similar analysis of information forces and potentials for the Shannon's entropy case could be carried out either using the original definition by Shannon or by considering the limit of the derivatives of Renyi's entropy approaching 1 from above and below.

3.3 Illustration of Information Forces

In this subsection, we will use two numerical examples to show the information forces and information potentials in single-dimensional and two-dimensional cases. In the first illustration, we consider the single-dimensional case with the kernel function chosen to be a Gaussian. In Figure 3-1, the one-dimensional information forces and information potential fields are shown for various kernel sizes. The attractive force field of an individual particle centered at the origin is plotted in Figure 3-1a. The forces can be made repulsive by introducing a negative sign in the definition. This procedure corresponds to choosing between minimizing or maximizing the sample entropy. Figure 3-1b shows the information potential at any point due to the existence of this particle at the origin as a function of distance to the origin. To further investigate the effect of additional samples on the potential and force fields, we position three additional randomly located samples. The final form of the overall quadratic information force field, which results as the superposition of the individual forces of these four particles, is shown in Figure 3-1c, and the overall quadratic information potential at a given location is presented as a function of position in Figure 3-1d. All plots include illustrations of their corresponding functions for various values of the selected kernel size. Notice that, as a consequence of the equivalence with sample variance we showed in Property 2.4, as the kernel size increases, the effective force becomes a linear function of distance, and is shown with the label *MSE* in Figure 3-1. For different kernel function choices, different force field definitions can be obtained, changing the adaptation dynamics.



(a)

(b)

(c)

(d)

Figure 3-1. Forces and potentials as a function of position for different values of kernel size a) force due to a single particle b) potential due to a single particle c) overall quadratic force at a given position due to all particles d) total quadratic potential at a given position

As a second illustration, we show a snapshot of a two-dimensional entropy maximization scenario, where the particles are bounded to within a unit square and interact under the quadratic force definition with Gaussian kernel choice. Since the objective is to maximize the entropy of the sample ensemble, the forces become repulsive. Given a set of randomly spaced samples in the unit square, when the forces acting on each sample are evaluated, it becomes evident that the information particles are pushed by the other particles in order to move along the direction of maximal entropy. The snapshot of the particle locations and the forces experienced by each particle is depicted in Figure 3-2.

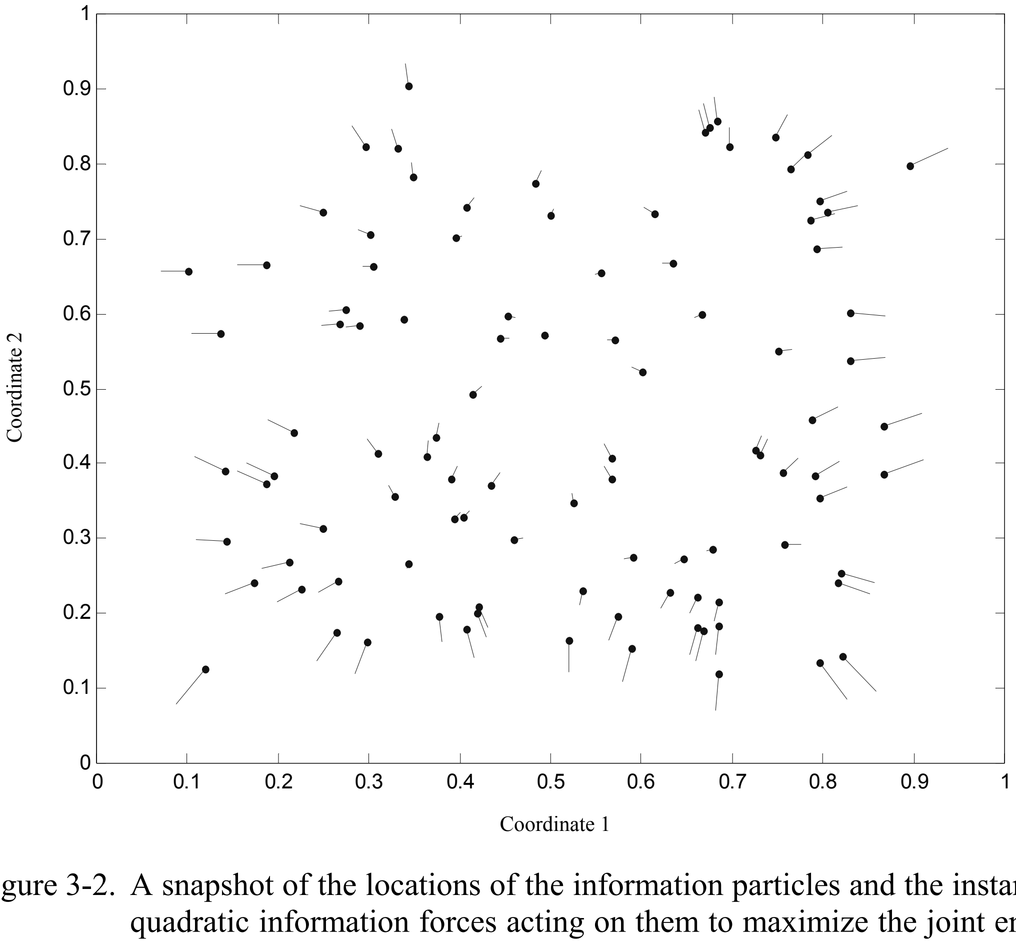


Figure 3-2. A snapshot of the locations of the information particles and the instantaneous quadratic information forces acting on them to maximize the joint entropy in the two-dimensional unit square

If the order-α information potentials were to be used, each information force shown in Figure 3-2, would have to be scaled with the corresponding factor that depends on the probability density of the particle and the parameter α.

3.4 Generalized Potential Energy and Particle Interaction Model for Learning

Traditionally, adaptation is regarded as an optimization process, where a suitable pre-defined performance criterion is maximized or minimized. Inspired by the above mentioned *information particle* idea, here we will propose an alternative view; we will treat each sample of the training data set as a particle and let these particles interact with each other according to the interaction laws that we define. The parameters of the adaptive system will then be modified in accordance with the interactions between the particles. Our aim is to determine a unifying model to describe the learning process as an interaction between *particles*, where the information particle model is a special case. We will, as well, show that even commonly used second-order statistics based adaptation criteria could be investigated under the same view of interacting particles.

3.4.1 Particle Interaction Model

Suppose we have the samples $\{z_1,\dots,z_N\}$ generated by some adaptive system. For simplicity, assume we are dealing with single dimensional random variables; however, note that extensions to multi-dimensional situations are trivial. In the particle interaction model, we assume that each sample is a particle and a potential field emanates from it. In this particle interaction model, *the value of each sample becomes the position of the corresponding particle*. In case of multi-dimensional sample vectors, this is the position vector in some coordinate frame. Suppose each particle z_i generates a potential energy field. Let this potential field be $v(\xi)$; we require this function to be continuous and differentiable, and to satisfy only the even symmetry condition $v(\xi) = v(-\xi)$, although physical potential fields (like the gravitational and electrical potentials) usually satisfy a spherical symmetry condition. Notice that due to the even symmetry and differentiability, the gradient of the potential function at the origin is zero (i.e., the force that a particle exerts on itself is zero). In addition, recall from physics that the force applied to a particle by another is usually a function of the relative position vector of the affected particle with respect to the source particle. With these in mind, we observe that the potential energy of particle z_j due to particle z_i, denoted by $V(z_j|z_i)$, is $V(z_j|z_i) = v(z_j - z_i)$, where the difference vector (z_j-z_i) gives the relative position of particle z_j with respect to z_i. The total potential energy of z_j due to all the particles in the training set is then given by

$$V(z_j) = \sum_{i=1}^{N} V(z_j \mid z_i) = \sum_{i=1}^{N} v(z_j - z_i) \tag{3.12}$$

We define the interaction force between these particles, in analogy to physics, as

$$F(z_j \mid z_i) \overset{\Delta}{=} \frac{\partial V(z_j \mid z_i)}{\partial z_j} = \left.\frac{\partial v(\xi)}{\partial \xi}\right|_{\xi=(z_j - z_i)} = v'(z_j - z_i) \tag{3.13}$$

From this and the superposition property of forces, we obtain the total force acting on particle z_j as

$$F(z_j) = \sum_{i=1}^{N} F(z_j \mid z_i) = \sum_{i=1}^{N} v'(z_j - z_i) \tag{3.14}$$

Notice that, as it should be, the force applied to a particle by itself is $F(z_j \mid z_j) = v'(0) = 0$. Finally, the total potential energy of the sample set is the sum (possibly weighted) of the individual potentials of each particle. Each particle could be weighed by a factor $\gamma(z_j)$ that may or may not depend on the particle's position. Assuming such a weighting, the total potential energy of the system of particles is found to be as given in Eq. (3.15).

$$V(z) = \sum_{j=1}^{N} \gamma \ (z_j) \sum_{i=1}^{N} v(z_j - z_i) \tag{3.15}$$

Assuming that $\gamma(z_j)=1$ for all samples (no weighting), we can determine the sensitivity of the overall potential of the particle system with respect to the position of a specific particle z_j. This is given by

$$\frac{\partial V(z)}{\partial z_k} = \frac{\partial}{\partial z_k} \sum_{j=1}^{N} \sum_{i=1}^{N} v(z_j - z_i) = \ldots = 2F(z_k) \tag{3.16}$$

In the adaptation context, where the samples are generated by a parametric adaptive system, the sensitivity of the total potential with respect to the weights of the system is also of interest. This sensitivity is directly related to the interaction forces between the samples as follows

$$\frac{\partial V}{\partial w} = \frac{\partial}{\partial w} \sum_{j=1}^{N} \sum_{i=1}^{N} v(z_j - z_i) = \ldots = \sum_{j=1}^{N} \sum_{i=1}^{N} F(z_j \mid z_i) \left(\frac{\partial z_j}{\partial w} - \frac{\partial z_i}{\partial w} \right) \tag{3.17}$$

3.4.2 Some Special Cases

Consider for example the potential function choice of $v(\xi) = \xi^2 / (2N^2)$ and weighting function choice of $\gamma \ (z_j) = 1$ (i.e., unweighted) for all samples. Then upon direct substitution of these values in Eq. (3.15), we obtain a $V(z)$ definition that equals the biased sample variance, i.e., minimization of this potential energy will yield the minimum variance solution for the weights of the adaptive system. In general, if we select potential functions of the form $v(\xi) = \left| \xi \ ^p \right|$, where $p>1$, with no weighting of the particles we obtain cost functions of the form

$$V(z)=\sum_{j=1}^{N}\sum_{i=1}^{N}\left|(z_j-z_i)^p\right| \tag{3.18}$$

which are directly related to the absolute central moments of the random variable *Z*, for which z_j's are samples. Each value of *p* corresponds to a different choice of the distance metric between the particles from the family of Minkowski norms.

The information potential we mentioned in the preceding sections also fall into this same category of energy functions, as they are the main inspiration for this generalization. Notice that for the potential function choice $v(\xi)=G_\sigma(\xi)/N^2$ and $\gamma\ (z_j)=1$ in Eq. (3.15), we obtain the quadratic information potential of Eq. (3.1). Additionally, for $v(\xi)=\kappa_\sigma(\xi)/N^2$ and $\gamma\ (z_j)=\hat{f}_Z^{\alpha-2}(z_j)$, we obtain Eq. (3.6) from Eq. (3.15). In the latter, we introduced the position-dependent weighting factor $\hat{f}_Z^{\alpha-2}(z_j)$ for each particle. The effect of this scaling was discussed in Section 3.2.

3.5 Backpropagation of Interaction Forces in MLP Training

In this section, we will derive the backpropagation algorithm for an MLP trained supervised under the *minimum energy learning* (MEL) principle; that is the adaptation of the MLP weights using a cost function of the form of Eq. (3.15), which is equivalent to adaptation according to the particle interactions defined by the forces in Eq. (3.13). This extended algorithm will backpropagate the interaction forces between the particles through the layers of the MLP instead of the error, as is the case in the standard MSE criterion case. For simplicity, consider the unweighted total potential of the error particles as the cost function. Assume that for multi-output situations, we simply sum the potentials of the error signals from each output.

Consider an MLP that has l layers with m_o processing elements (PE) in the o^{th} layer. We denote the input vector with layer index zero. Let w^o_{ji} be the weight connecting the i^{th} input to the j^{th} output in the o^{th} layer. Let $v^o_j(s)$ be the synapse potential of the j^{th} PE at o^{th} layer corresponding to the input sample $x(s)$, where s is the sample index. Let $\varphi(.)$ be the sigmoid nonlinearity of the MLP, same for all PEs, including the output layer. Assume $v(\xi)$ is the potential function of choice and we have N training samples. The total energy of the system of error particles, given by $\{e(1),\ldots,e(N)\}$, is then

$$V=\sum_{s=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{m_l}v(e_k(s)-e_k(t))\overset{\Delta}{=}\sum_{s=1}^{N}\sum_{t=1}^{N}\varepsilon(s\,|\,t) \tag{3.19}$$

The derivation of the *backpropagation of interaction forces* algorithm follows similar lines to that of the conventional error backpropagation algorithm [Hay99, Rum86]. The total potential energy of the output errors summed over the output PEs, for a given sample pair $(s|t)$ is defined as

$$\varepsilon(s\,|\,t)\overset{\Delta}{=}\sum_{k=1}^{m_l}v(e_k(s)-e_k(t)) \tag{3.20}$$

For this MLP, the k^{th} outputs before and after the nonlinearity of the last layer are respectively given by

$$\begin{aligned} v^l_k &= \sum_{i=0}^{m_{l-1}} w^l_{ki}\, y^{l-1}_i \\ y^l_k &= \varphi(v^l_k) \end{aligned} \tag{3.21}$$

Taking the derivative of $\varepsilon(s\,|\,t)$ with respect to the output layer weights, we obtain Eq. (3.22), where $\varphi'(.)$ is the derivative of the MLP's sigmoid function and $\delta^l_k(.|.)$ are the sensitivities of the local energy potentials in the network that depend on the interaction forces between the indicated particles; notice that their definitions include $F(s\,|\,t) = v'(e(s)-e(t))$, i.e., the interaction forces.

$$\begin{aligned}
\frac{\partial \varepsilon(s\,|\,t)}{\partial w_{ki}^{l}} &= v'(e(s)-e(t))\cdot\left[-\varphi'(v_k^l(s))y_i^{l-1}(s)+\varphi'(v_k^l(t))y_i^{l-1}(t)\right] \\
&= v'(e(s)-e(t))\varphi'(v_k^l(t))y_i^{l-1}(t)-v'(e(s)-e(t))\varphi'(v_k^l(s))y_i^{l-1}(s) \\
&= -v'(e(t)-e(s))\varphi'(v_k^l(t))y_i^{l-1}(t)-v'(e(s)-e(t))\varphi'(v_k^l(s))y_i^{l-1}(s) \\
&\overset{\Delta}{=} \delta_k^l(t\,|\,s)y_i^{l-1}(t)+\delta_k^l(s\,|\,t)y_i^{l-1}(s)
\end{aligned} \tag{3.22}$$

For the hidden node *l*-1, we can write, similarly,

$$\begin{aligned}
\frac{\partial \varepsilon(s\,|\,t)}{\partial w_{ki}^{l-1}} &= v'(e(s)-e(t))\left[\begin{array}{l} \dfrac{\partial e(s)}{\partial y_j^{l-1}(s)}\dfrac{\partial y_j^{l-1}(s)}{\partial v_j^{l-1}(s)}\dfrac{\partial v_j^{l-1}(s)}{\partial w_{ji}^{l-1}} \\ -\dfrac{\partial e(t)}{\partial y_j^{l-1}(t)}\dfrac{\partial y_j^{l-1}(t)}{\partial v_j^{l-1}(t)}\dfrac{\partial v_j^{l-1}(t)}{\partial w_{ji}^{l-1}} \end{array}\right] \\
&= v'(e(s)-e(t))\left[\begin{array}{l} -\sum_{k=1}^{m_l}\varphi'(v_k^l(s))w_{kj}^l\varphi'(v_j^{l-1}(s))y_i^{l-2}(s) \\ +\sum_{k=1}^{m_l}\varphi'(v_k^l(t))w_{kj}^l\varphi'(v_j^{l-1}(t))y_i^{l-2}(t) \end{array}\right] \\
&= \sum_{k=1}^{m_l}\delta_k^l(t\,|\,s)w_{kj}^l\varphi'(v_j^{l-1}(t))y_i^{l-2}(t) \\
&\quad + \sum_{k=1}^{m_l}\delta_k^l(s\,|\,t)w_{kj}^l\varphi'(v_j^{l-1}(s))y_i^{l-2}(s) \\
&\overset{\Delta}{=} \delta_k^{l-1}(t\,|\,s)y_i^{l-2}(t)+\delta_k^{l-1}(s\,|\,t)y_i^{l-2}(s)
\end{aligned} \tag{3.23}$$

The sensitivities of the other hidden layers (if there are more than two) can be computed using the same idea, resulting in similar equations. This derivation, and the main points of the algorithm can be summarized as follows. In the algorithm below, η is the learning rate. Notice that for $v(\xi)=\xi^2$, the algorithm reduces to the backpropagation of error, since the force becomes $F(e_j(s)\,|\,e_j(t)) = 2(e_j(s)-e_j(t))$.

Algorithm 3.1. Let the interaction force acting on sample *s* due to the potential field of sample *t* be $F(e_j(s)\,|\,e_j(t)) = v'(e_j(s) - e_j(t))$ in the *j*th output node of the MLP. These interactions will minimize the energy function in Eq. (3.19).

1. Evaluate local gradients for the output layer for *s*,*t*=1,…,*N* and *j*=1,…,*m*$_l$ using

$$\begin{aligned}\delta_j^l(s\,|\,t) &= -F(e_j(s)\,|\,e_j(t))\cdot\varphi'(v_j^l(s))\\ \delta_j^l(t\,|\,s) &= -F(e_j(t)\,|\,e_j(s))\cdot\varphi'(v_j^l(t))\end{aligned} \tag{3.24}$$

2. For layer index *o* going down from *l*-1 to 1 evaluate the local gradients

$$\begin{aligned}\delta_j^o(s\,|\,t) &= \varphi'(v_j^o(s))\sum_{k=1}^{m_{o+1}}\delta_k^{o+1}(s\,|\,t)w_{kj}^{o+1}\\ \delta_j^o(t\,|\,s) &= \varphi'(v_j^o(t))\sum_{k=1}^{m_{o+1}}\delta_k^{o+1}(t\,|\,s)w_{kj}^{o+1}\end{aligned} \tag{3.25}$$

3. For each layer index *o* from 1 to *l* evaluate the weight updates (to minimize V)

$$\Delta w_{ji}^o = -\eta\,\left(\delta_j^o(s\,|\,t)y_i^{o-1}(s) + \delta_j^o(t\,|\,s)y_i^{o-1}(t)\right) \tag{3.26}$$

The energy potential cost functions, in general are insensitive to the mean position of the particles, therefore, in applications where the mean of the samples is also desired to approach a certain value (for example to zero in supervised training), an external force acting on all the particles to draw them in that direction may be introduced. In that case, the interaction forces in the definition of the last layer sensitivities in Eq. (3.24) must be replaced by the superposition of the interaction force and the external force acting on that particle. In the physical analogy, this additional force can be viewed as an external (other than the particles themselves) effect.

Adaptive systems research is traditionally motivated by the optimization of suitable cost functions and is centered on the investigation of learning algorithms that achieve the desired optimal solution. In this section, inspired by the idea of *information theoretic learning through particle interactions* introduced by Principe *et al.* [Pri00a] and expound in Section 3.2, we proposed an alternative approach to adaptation and learning. This new approach allows us to regard the adaptation process in analogy with interacting particles in a force field (also generated by the same particles) in physics. Besides the intellectual appeal of this viewpoint provides us for further theoretical study on learning, it might be promising in designing real systems that use physical forces to change its state and eventually adapt to its environment to need.

CHAPTER 4
MINIMUM ERROR ENTROPY CRITERION FOR SUPERVISED LEARNING

4.1 Minimum Error Entropy Criterion

Supervised learning algorithms traditionally use the MSE criterion as the figure of merit, which is a sufficient statistics for the case of linear systems under Gaussian residual error assumptions as in the work of Wiener [Far98, Hay84, Wie49]. Although the Gaussianity assumption, which is supported by central limit theorem, and second-order statistics provide successful engineering solutions to most practical problems, it has become evident that when dealing with nonlinear systems, this approach needs to be refined [Pri00a]. Therefore, criteria that not only consider the second-order statistics but that also take into account the higher-order statistical behavior of the systems and signals are desired. Recent papers addressed this issue both in the control literature [Fen97] and the signal processing / machine learning literature [Cas00, Erd00a, Fis00]. In a statistical learning sense, especially for nonlinear signal processing, a more appropriate approach would be to constraint directly the information content of signals rather than simply their energy, if the designer seeks to achieve the best performance in terms of information filtering [Dec96, Kap92, Lin88].

Since entropy is defined as the average information content in a random variable, it is only natural to adopt it as the criterion for applications where manipulation of the information content of signals is desired or necessary. In fact, the entropy criterion can generally be used as an alternative for MSE in supervised adaptation [Hay98]

The goal in dynamic modeling is to identify the nonlinear mapping that produced the given input-output data. This is traditionally achieved in a predictive framework as shown in Figure 4-1 [Hay99].

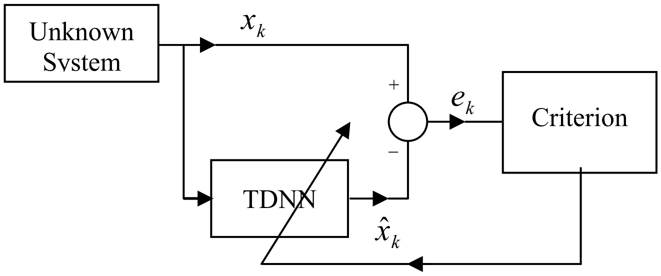


Figure 4-1. Time-delay neural network prediction scheme

Minimization of MSE in the criterion block simply constrains the square difference between the original trajectory and the trajectory created by the adaptive system (TDNN in this case), which does not guarantee the capturing of all the information about the underlying dynamics. Hence, we propose here the minimization of the error entropy (MEE) as a more robust criterion for dynamic modeling, and an alternative to MSE in other supervised learning applications using nonlinear systems, such as nonlinear system identification with neural networks [Erd02c].

The intuition behind the entropy criterion for supervised learning is conceptually straightforward: Given samples from an input-output mapping, in order to extract the most information from the data, the information content of the error signal must be minimized; hence the error entropy over the training data set must be minimized. In this chapter, we will show that minimizing the error entropy is equivalent to minimizing the Renyi's divergence between the probability distributions of the desired and system outputs. These distance measures, from an information-geometry view point, are directly related to the divergence of the statistical models in probability spaces [Ama85].

As for the entropy, we will choose Renyi's definition and use our nonparametric estimator as a substitute in finite-sample training scenarios. However, before proceeding with the application examples of MEE in supervised training scenarios, we investigate some mathematical properties of it.

4.2 Properties of Minimum Error Entropy Criterion

In Chapter 2, when investigating the general properties of the entropy an our estimator, we saw that the entropy is invariant under changes in the mean value of the pdf. This is not a problem, however. For example, when training a linear-output-layer MLP, once the training of all the other weights are completed according to the MEE criterion, we can set the output bias of the MLP to match the sample average of the MLP output to the sample average of the desired output. In addition, we showed in the same section that as the kernel size in the estimator is increased, the entropy estimate approaches to the log of a scaled version of the sample variance, thus in this special case minimization of error entropy and error variance become identical asymptotically. Also we know that the global minimum of the kernel estimator occurs when all the samples identically assume the same value, just what we want for the error samples in supervised training – all zeros. The conditions of symmetry and differentiability for the kernel function for this to occur, however, must be noted when designing the cost function and the learning algorithm.

Theorem 4.1**.** Minimizing Renyi's error entropy minimizes the Renyi's divergence between the joint pdfs of the input-desired signals and the input-output signals. In the special case of Shannon's entropy, this reduces to minimizing the K-L divergence [Erd02d].

Proof. The error is given as the difference between the desired output and the actual output, i.e., $e = d - y$. Using this identity, we can relate the pdf of error to the pdf of the output as $f_{e,w}(e) = f_{y|x,w}(d - e \mid x)$, where the subscript w denotes dependence on the optimization parameters, the weight vector of the adaptive system. Minimum error entropy problem is formulated as follows.

$$\begin{aligned} \min_w \frac{1}{1-\alpha} \log \int_{-\infty}^{\infty} f_{e,w}^{\alpha}(e)de &= \frac{1}{1-\alpha} \log \int_{-\infty}^{\infty} f_{y|x,w}^{\alpha}(d - e \mid x)de \\ &= \frac{1}{1-\alpha} \log \int_{-\infty}^{\infty} - f_{y|x,w}^{\alpha}(y \mid x)dy \qquad \text{(variable change } y = d - e) \end{aligned} \tag{4.1}$$

Consider the case $\alpha > 1$. From now on, we will drop the limits from the integral signs for convenience of typing.

$$\begin{aligned} &\equiv \min_w \int f_{y|x,w}^{\alpha}(y \mid x)dy \cdot \int f_x^{\alpha}(x)dx = \iint f_{xy,w}^{\alpha}(x,y)dxdy \equiv \\ &\equiv \iint f_{xy,w}^{\alpha}(x,y)dxdy \cdot \iint f_{xd}^{1-\alpha}(x,y)dxdy \\ &= \min_w \iint f_{xy,w}(x,y) \left(\frac{f_{xd}(x,y)}{f_{xy,w}(x,y)} \right)^{1-\alpha} dxdy \end{aligned} \tag{4.2}$$

We recognize this final expression as the argument of the log in the Renyi's divergence measure. Inserting the missing log and the scale factor that depends on the entropy order, which does not change the minimization problem since $\alpha > 1$ and log is monotonic, we obtain

$$\min_w \frac{1}{\alpha - 1} \log \iint f_{xy,w}(x,y) \left(\frac{f_{xd}(x,y)}{f_{xy,w}(x,y)} \right)^{1-\alpha} dxdy \tag{4.3}$$

A similar derivation may be carried out for the case $\alpha < 1$, yielding the same result. As for the reduction to K-L divergence, taking the limit of Eq. (4.3) as $\alpha \to 1$

using L'Hopital's rule produces the desired result. In fact, starting the derivation from Shannon's entropy definition for the error, one arrives directly at the K-L divergence. For interested readers, this derivation is provided in Appendix A.

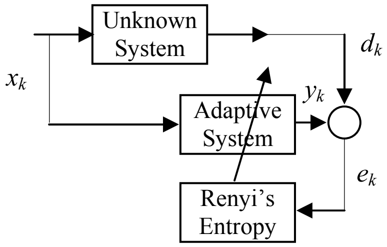


Figure 4-2. Schematic diagram of supervised learning using Renyi's error entropy as the performance index

Theorem 4.2. Minimum error entropy criterion is asymptotically robust to additive zero-mean noise, regardless of the pdf of this noise.

Proof. Consider the learning process depicted in Figure 4-2. Suppose that the desired signal consists of the superposition of a deterministic part and a zero-mean random part, such that $d = g(x) + v$, where $g(.)$ is the unknown function that the adaptive system is trying to identify and v is the zero-mean noise with pdf $f_v(.)$ independent from x, d, and y. Suppose the learning system is a parametric family of functions of the form $h(x;w)$ where w is the vector of parameters, called the weight vector. Assume x, d, and y are all zero-mean signals without loss of generality. Let w_* be (one of possibly many) optimal weight vectors than minimize the error entropy, where the error signal is defined as e=d-y. Let $\overline{w}_*$ be the optimal weight vector that minimizes the entropy of the *clean* error signal that is defined as $\overline{e} = g(x) - h(x,w)$. Notice that we have the identity $e = \overline{e} + v$. Since v is an independent noise signal that does not depend on w, the weights of the adaptive system, when $\overline{e}$ is δ- distributed we have

$$w_* = \arg\min_w H_\alpha(e(w)) = \arg\min_w H_\alpha(\overline{e}(w) + v) = \arg\min_w H_\alpha(\overline{e}(w)) = \overline{w}_* \qquad (4.4)$$

Even if $\bar{e}$ is not δ - distributed (which occurs when the model span does not include the actual system), due to Theorem 2.4, we can argue that, in general, $H_\alpha(\bar{e}+v) \geq H_\alpha(\bar{e})$ and minimizing this upper bound may force the solution to converge to a good value, which would be obtained in the noise-free situation. An alternative proof, which provides better insights about the process, is in Appendix B.

Another property of the entropy estimator, which relates directly to the algorithm performance in supervised adaptation with MEE is its close relationship with the method of convolutional smoothing in global optimization [Rub81]. Convolution smoothing was proven effective in practical applications; for instance consider the adaptation of IIR filters [Edm96]. The basic idea behind this approach is to convolve the cost function with a wide *smoothing functional*, which eliminates the local minima initially. The width of the smoothing functional can then be gradually decreased until a Dirac-δ is obtained, which leaves the original cost function. During this course, the optimization parameters come to the vicinity of the global optimum and are in the domain of attraction for this solution. We can briefly describe the method and the requirements as follows.

The global convergence theorem for convolution smoothing states that the following optimization problems are equivalent

$$\min_{x \in D \subset \Re^n} g(x) = g(x^*) = \min_{x \in D \subset \Re^n} \hat{g}_\beta(x), \quad \beta \to 0 \tag{4.5}$$

where the smoothened cost function is defined as

$$\hat{g}_\beta(x) \overset{\Delta}{=} g(x) * h_\beta(x) \tag{4.6}$$

and thus both problems result in the global optimal point x^* [Rub81]. There are conditions that the smoothing functional $h_\beta(x)$ has to satisfy.

i. $h_{\beta}(x) = (1/\beta^n)h(x/\beta)$ (4.7)

ii. $\lim\limits_{\beta \to 0} h_{\beta}(x) = \delta(x)$ (4.8)

iii. $\lim\limits_{\beta \to 0} \hat{g}_{\beta}(x) = g(x)$ (4.9)

iv. $h_{\beta}(x)$ is a pdf (4.10)

Condition (iii) guarantees that both $g(x)$ and $h_{\beta}(x)$ are well-behaved functions. Condition (iv) allows the proof techniques from the stochastic optimization literature to be applicable. For our purposes, this strict condition is not a necessary, since even if the convolving function does not integrate to one, then the same convolution smoothing effect will be observed, except there will be a scale factor that multiplies the smoothed functional. The most important constraints on the smoothing function are (i) and (ii).

Conjecture 4.1. Given a specific choice of the kernel function $\kappa_{\sigma}(.)$, there exists a corresponding smoothing functional $h_{\beta}(.)$, which is a solution of

$$\overline{V}_{\alpha,\sigma}(w) = V_{\alpha}(w) * h_{\beta}(w) \tag{4.11}$$

where $V_{\alpha}(w) = \int f_e^{\alpha}(e;w)de$, $\overline{V}_{\alpha,\sigma}(w) = \lim\limits_{N \to \infty} \hat{V}_{\alpha,\sigma}(e)$ and that satisfies the conditions (i)-(iv) given above.

Support: There are a number of theoretical and experimental observations that support this conjecture. First, consider the nonparametric information potential we use for the error samples.

$$\hat{V}_{\alpha,\sigma}(e) = \frac{1}{N^{\alpha}} \sum_j \left(\sum_i \frac{1}{\sigma} \kappa_{\sigma}\left(\frac{e_j - e_i}{\sigma} \right) \right)^{\alpha-1} = \frac{1}{\sigma^{\alpha-1}} \hat{V}_{\alpha,1}(e/\sigma) \tag{4.12}$$

Notice that the change in kernel size causes dilation in the *e*-space. Therefore, all points, including all local extremes move radially away from the origin when σ is increased. The only point that maintains its position is the origin. From this, we conclude that if the span of the function approximator (adaptive system) that is being used covers the function being approximated (i.e., the error at the optimal point is zero), then the location of the global solution is independent of the kernel size. Also, if the function approximator used is a contractive mapping, which is the case in feedforward neural networks for example, then the dilation in the *e*-space must be followed by dilation in the weight-space, hence the volume of the domain of attraction of the global optimum is increased.

In addition, consider the asymptotic behavior of the nonparametric information potential estimator. Since Parzen windowing is a consistent estimator, as the number of samples goes to infinity, the estimated pdf converges to the actual pdf convolved with the kernel function that is used [Par67], which also happens in the mean as we mentioned before.

$$\overline{V}_{\alpha,\sigma}(w) = \int [f_e(e;w) \underset{e}{*} \kappa_\sigma(e)]^\alpha \, de \tag{4.13}$$

where $\underset{e}{*}$ denotes a convolution with respect to the variable *e*. Equating (4.13) to the convolution of the true information potential $V_\alpha(w)$ and the (hypothetical) smoothing functional $h_\beta(w)$, we obtain the condition Eq. (4.11). Consider the explicit form of this equality written in terms of the kernel function and the error pdf.

$$h_\beta(w) \underset{w}{*} \int f_e^\alpha(e;w) de = \int \left[f_e\ (e;w) \underset{e}{*} \kappa_\sigma(e) \right]^\alpha de \tag{4.14}$$

Taking the Laplace transform of both sides with respect to *w*, we can isolate the Laplace transform of $h_\beta(w)$ in terms of the transforms of the remaining quantities. The Laplace transform of $h_\beta(w)$ is guaranteed to exist if the error pdf and the kernel function are absolutely integrable functions and $\alpha \geq 1$, which is the case. We can write this function in the transform domain as the following ratio.

$$H_\beta(s) = \frac{L_w \int [f_e(e;w) \underset{e}{*} \kappa_\sigma(e)]^\alpha de}{L_w \int f_e^\alpha(e;w) de} = \frac{\int L_w [f_e(e;w) \underset{e}{*} \kappa_\sigma(e)]^\alpha de}{\int L_w [f_e^\alpha(e;w)] de} \tag{4.15}$$

The right-hand side is a function of *s* only, since the integration over *e* from $-\infty$ to ∞ eliminates this variable.

Since $H_\beta(s)$ exists, $h_\beta(w)$ must be absolutely integrable, therefore, $\lim_{w\to\pm\infty} h_\beta(w) = 0$. We next observe that as $\sigma \to 0$, the numerator of Eq. (4.15) converges to the denominator, hence the bandwidth of $H_\beta(\omega)$ (considering the Fourier transform) increases. An increase in frequency-domain bandwidth is accompanied by a decrease in duration of the impulse response in time-domain, thus the width of $h_\beta(w)$ decreases as $\sigma \to 0$; note that there is a nonlinear monotonous relation between β and σ.

Now that we know the width of $h_\beta(w)$ decreases monotonously as $\sigma \to 0$, that it is always absolutely integrable, and that it converges to $\delta(w)$ in the limit, we conclude that it has to be unimodal, symmetric, and positive for all w. Consequently, even if $h_\beta(w)$ does not integrate to 1, it integrates to some finite value and therefore it is a scaled version of a pdf. A scaling factor in the convolution process does not affect the nature of the smoothing but only the scale factor of the smoothed performance surface.

Although it is not easy to solve for the corresponding smoothing function from Eq. (4.11), we showed that the solution still satisfies some of the required conditions, specifically (ii)-(iv). Furthermore, the dilation in the *e*-space, presented in Eq. (4.12), hints towards the validity of condition (i). However, it has not been possible to verify that the first condition is satisfied in general for any mapper, nor it was possible to set forth the conditions under which this occurs. Therefore, we propose the existence of a smoothing functional corresponding to each kernel choice as a conjecture. Consequently, we propose the following methodology to achieve global optimization when using the current nonparametric entropy estimator in MEE-training of adaptive systems: Start with a large kernel size, and during the adaptation gradually and slowly decrease it towards a predetermined suitable value; the local solutions, which would trap the training for those same initial conditions when the *w*-space has not been dilated, will be avoided. Hence, global optimization will be achieved still using a gradient descent approach.

4.3 Simulation Examples on Minimum Error Entropy Criterion

In the preceding sections, we introduced the concept of MEE training for adaptive systems, applicable in supervised learning schemes and investigated some of the mathematical properties of the criterion itself and the properties arising from the structure of the entropy estimator that we use. In this section, we provide the gradient expression for this cost function and present simulation results from numerous applications of this gradient in steepest descent training of nonlinear adaptive systems, mainly MLPs.

Suppose the error samples in a supervised training scenario are generated according to the equation $e_k = d_k - y_k$, where k is the sample index and d_k and y_k are respectively the desired and the actual adaptive system outputs. Let the output of the adaptive system be defined in terms of its input vector x_k as $y_k = g(x_k;w)$, where w denotes the dependence of the input-output mapping on the system weight vector. Since our aim is to minimize the error entropy, for α>1, we can alternatively maximize the order-α information potential. Using our nonparametric estimator for this quantity on the N error samples, we can determine the gradient of the information potential of the error with respect to the weights of this adaptive system as

$$\frac{\partial \hat{V}_\alpha(e)}{\partial w} = -\frac{(\alpha-1)}{N^\alpha}\sum_j\left(\sum_i \kappa_\sigma(e_j-e_i)\right)^{\alpha-2}\left[\sum_i \kappa'_\sigma(e_j-e_i)\left(\frac{\partial y_j}{\partial w}-\frac{\partial y_i}{\partial w}\right)\right] \tag{4.16}$$

where all summations are over the sample index from 1 to N. The above gradient can be used for single-output supervised training scenarios. For the case of multiple outputs, the cost function may be modified as the sum of marginal error entropies of each outputs, or alternatively, the product of the individual information potentials. For a system with d outputs, for instance, the corresponding cost function would be

$$\min_w J(w) = \min_w \sum_{o=1}^n H_\alpha(e^o) \overset{\alpha>1}{\equiv} \max_w \prod_{o=1}^n V_\alpha(e^o) \tag{4.17}$$

where e^o denotes the error signal for the o^{th} output of the adaptive system. With this cost function, then the gradient has to be modified to

$$\frac{\partial J}{\partial w} = \sum_{o=1}^n \frac{1}{1-\alpha}\frac{\partial V_\alpha(e^o)/\partial w}{V_\alpha(e^o)} \quad or \quad \sum_{o=1}^n\left(\prod_{p\neq o} V_\alpha(e^p)\right)\frac{\partial V_\alpha(e^o)}{\partial w} \tag{4.17}$$

A second approach for multi-output situations is to minimize the joint entropy of the error vector, however, as we know the data requirements for accurately estimating a statistical quantity in high-dimensional data spaces, in general, requires an exponentially increasing number of samples.

4.3.1 Prediction of the Mackey-Glass Chaotic Time Series

Our first example is the single-step prediction of the well-known Mackey-Glass chaotic time series, which often serves as a benchmark data set in testing prediction algorithms in the literature, yet recently more difficult-to-predict time series like the Lorenz series is becoming popular [Cas01, Dur99, Xiq99]. The Mackey-Glass series has a delay-based chaotic behavior and the attractor associated with the given delay amount is usually denoted with that number [Kap95]. For our simulations, we will use samples drawn at *T*=1sec intervals from the MG30 attractor whose continuous time dynamics are defined by the following continuous-time differential equation. The integration is performed using the Runge-Kutta4 method with time-step equal to 0.1sec, and then the generated series was down sampled by 10, to get the desired sampling period of 1sec.

$$\dot{x}(t) = -0.1x(t) + \frac{0.2x(t-30)}{1+x(t-30)^{10}} \tag{4.16}$$

In all the following simulations regarding the MG30 data, we used 200 samples for training and the 10000 test samples are generated using a different initial condition, thus are from a different trajectory on the same attractor.

As the aim of our first set of simulations is to compare the generalization properties of learning with MSE versus learning with MEE, we train two different sets of MLPs on the same data; one of these groups uses MSE as the criterion and the other uses MEE. In addition, in order to make sure that the results we obtain are not dependent on the specific TDNN architecture we choose and its capabilities, we include 8 different 2-layer TDNNs in each group whose number of hidden neurons vary from 3 to 10. To increase the speed of training for all 16 TDNNs we use the conjugate gradient approach [Lue73]. However, in need of avoiding local optimum solutions we take the Monte Carlo approach to select the initial conditions for the weight vectors and use 1000 (uniformly distributed) randomly selected sets of weights for each TDNN. After all 16 TDNNs are trained starting from all these 1000 initial weight vectors, the optimal weight vectors for those TDNNs trained according to the MSE criterion were selected to be the solutions obtained from among the 1000 different runs that yield the smallest MSE and similarly the optimal weight vectors for those TDNNs trained according to the MEE criterion were selected as those that yield the smallest error entropy. Afterwards, these solutions were iterated a couple of more epochs in order to test and to guarantee their convergence to the minimum; in fact, visual inspection of the learning curves for all TDNNs showed that with the conjugate gradient approach, all TDNNs using the MSE criterion converged in less than 100 iterations and all TDNNs using the MEE criterion converged in less than 30 iterations. It must be noted, however, that the computational complexity of the gradient of entropy is greater than that of the squared error. The kernel function used to estimate the entropy in all simulations was set to a Gaussian with size $\sigma = 0.01$ experimentally. In addition, the output bias of the linear output neurons are set to match the sample mean of the system output to that of the desired output for both MSE and MEE criteria. In all these first set of simulations, we use Renyi's quadratic entropy definition.

Our first comparison is between the central moments of the desired and predicted MG30 values on the test data [Erd02c]. These results are summarized in Figure 4-3. Remember that the MSE trained TDNNs minimize the error variance, which is the 2nd central moment, on the training data, however, there is no guarantee that they will exhibit smaller variance than the MEE trained TDNNs on the test set. In fact, this is exactly what we observe in Figure 4-3. The MEE criterion achieves smaller variance for all sizes of the network except for 6 hidden neurons. For this reason, in the following, we elaborate on this special case, where the TDNN has 6 hidden PEs.

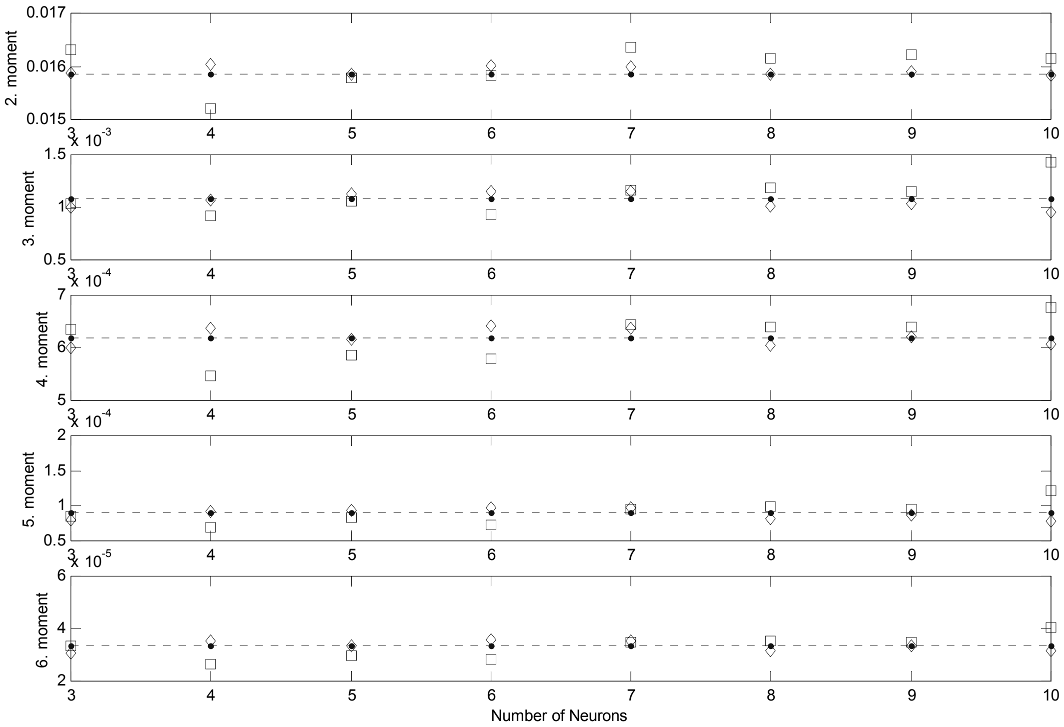


Figure 4-3. Central moments up to order 6 of the desired output (dashed), the predicted MG30 series for MEE-trained (diamonds) and MSE-trained (squares) TDNNs versus the number of hidden neurons

Figure 4-4 shows the performances of MSE and MEE criteria in capturing the underlying pdf behind the desired signal. Clearly, the TDNN that is trained using the minimum entropy principle extracted more information regarding this objective and achieved a better model that represents the underlying statistical structure of the data. This is expected as we showed that the MEE principle equivalently tries to match the pdf of the adaptive system output to that of the desired signal in the Renyi's divergence sense, where as the MSE criterion merely tries to estimate the conditional expectation of the desired output given the input [Bis95].

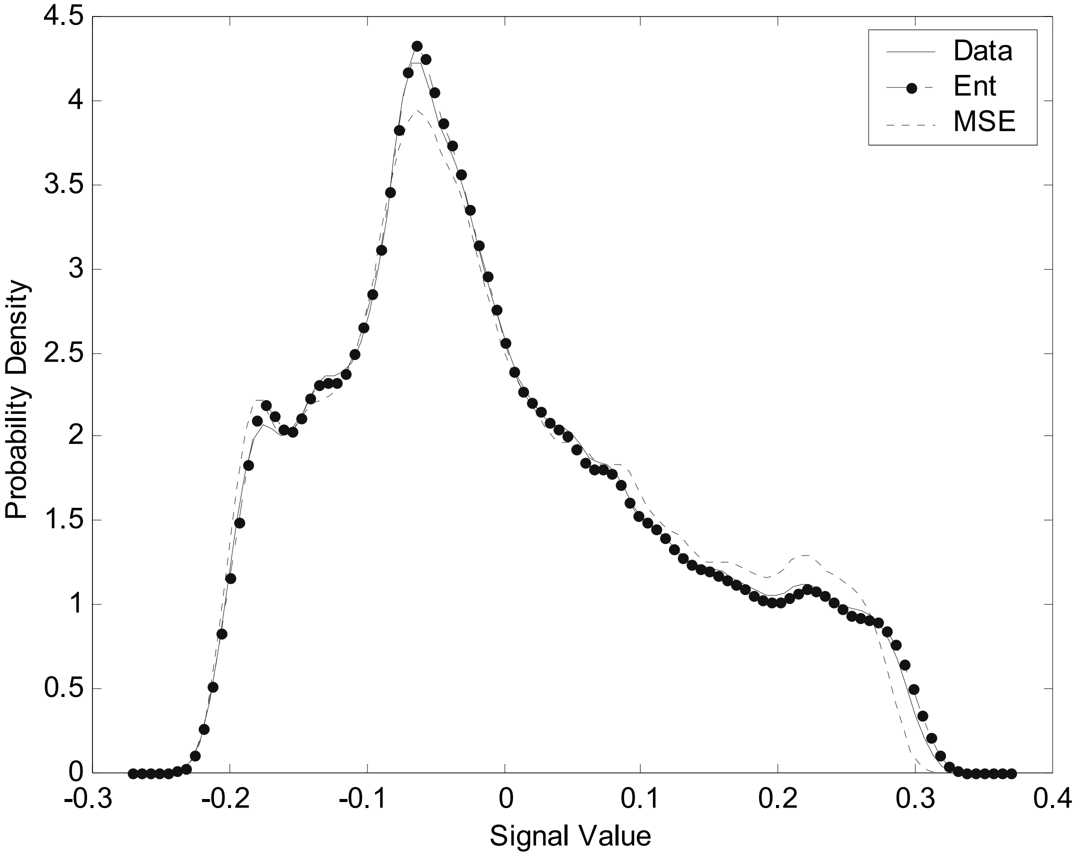


Figure 4-4. Probability density estimates of the 10000-sample MG30 test series (solid) and its predictions by MEE-trained (thick dots) and MSE-trained (dotted) TDNNs. All pdfs are normalized to zero-mean

Our second set of simulations is aimed to investigate the effect of entropy order on the performance of the final solution obtained. The effect of the kernel size is studied as well. For each set of parameters (kernel size and entropy order) we run 100 Monte Carlo runs using randomly selected initial weight vectors. At the end of the training, which used 200 samples, the information potential of the error on the test set consisting of 10000 samples corresponding to each TDNN was evaluated using a Gaussian kernel of size $\sigma = 10^{-3}$ to provide a basis for fair comparison. For the final error signals obtained, this value of the kernel size allows the kernels to cover about 10 samples on the average, which is a rule of thumb that has been determined at CNEL and experimentally verified by myself for accurate information potential estimations (verbal communication with Principe). The results are summarized in Table 4-1 in the form of normalized information potentials [Erd02d]. There are a total of twelve trained TDNNs, using the designated entropy orders and kernel sizes given in the first column. The performances of these TDNNs are then evaluated and compared using four different entropy orders, presented in each column. When inspecting these results, the data in each column should be compared. Each row corresponds to the performance of the TDNN trained using the parameter values designated with each column giving the evaluation of the information potential for this data using different entropy orders mentioned in the first row of that column. Notice that, regardless of the entropy order used in evaluation (each column), the TDNN trained using the quadratic entropy (α = 2) yields the best performance. Furthermore, using smaller kernel sizes in training also improved the final performance.

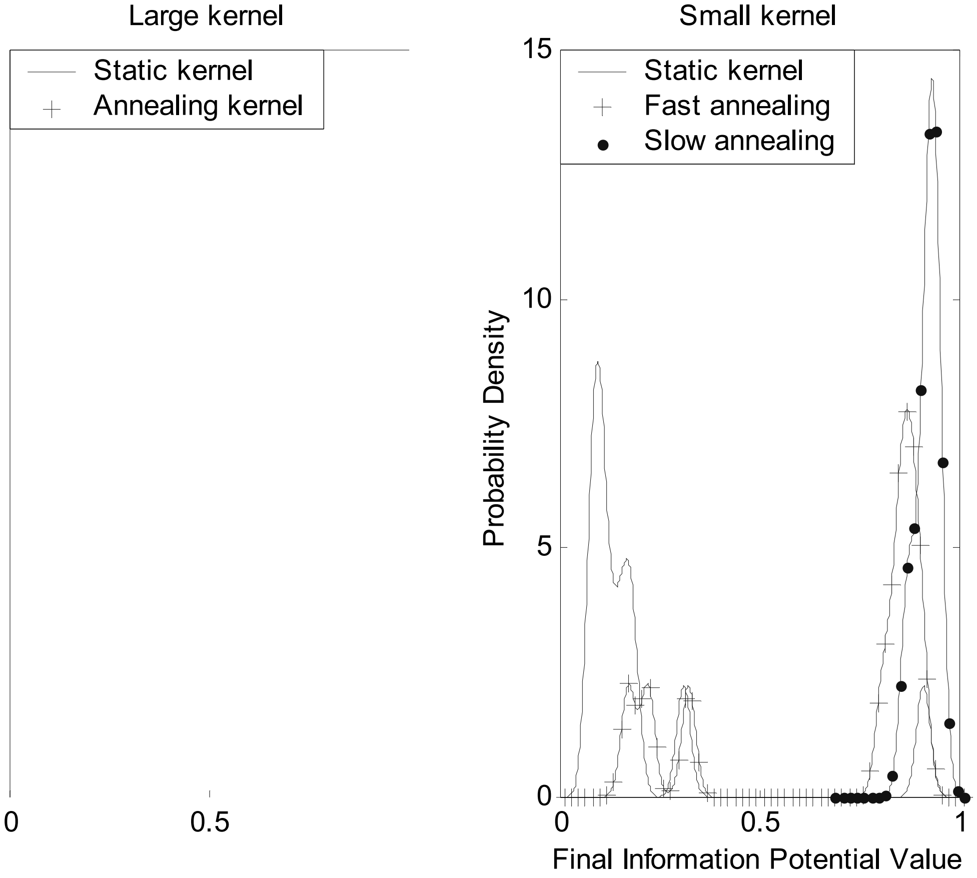
Table 4-1. Evaluation of the normalized information potential at different entropy orders of the error samples for TDNNs trained with different parameters. The normalization procedure consists of dividing by the maximum possible theoretical value

Training parameters	Evaluation parameters	$V_\alpha(e)$ $\alpha = 1.01$ $\sigma = 10^{-3}$	$V_\alpha(e)$ $\alpha = 1.5$ $\sigma = 10^{-3}$	$V_\alpha(e)$ $\alpha = 2$ $\sigma = 10^{-3}$	$\mathrm{V}_\mathrm{n}(\mathrm{e})$ $\alpha = 3$ $\sigma = 10^{-3}$
$\alpha = 1.01$	$\sigma = 0.01$	0.976	0.304	0.099	0.012
	$\sigma = 0.1$	0.976	0.311	0.104	0.013
	$\sigma = 1$	0.969	0.212	0.047	0.002
$\alpha = 1.5$	$\sigma = 0.01$	0.977	0.321	0.112	0.016
	$\sigma = 0.1$	0.977	0.318	0.109	0.015
	$\sigma = 1$	0.976	0.312	0.105	0.014
$\alpha = 2$	$\sigma = 0.01$	**0.979**	**0.352**	**0.135**	**0.023**
	$\sigma = 0.1$	**0.979**	**0.352**	**0.133**	**0.021**
	$\sigma = 1$	**0.978**	**0.343**	**0.126**	**0.019**
$\alpha = 3$	$\sigma = 0.01$	0.977	0.336	0.124	0.020
	$\sigma = 0.1$	0.977	0.330	0.117	0.017
	$\sigma = 1$	0.976	0.312	0.105	0.014

Our third set of simulations investigates the validity of the conjecture on the global optimization capabilities of our MEE algorithm. In these simulations, we use the quadratic entropy criterion on the MG30 training data again. This time, however, the size of the Gaussian kernels is annealed during the training from a large value to a smaller one. Once again the Monte Carlo approach is taken with 100 randomly selected initial weight vector assignments.

The results of these experiments are summarized in Figure 4-5 as the pdf estimates of the final normalized information potential values (so that the maximum value is one) obtained on the training data [Erd02d]. In Figure 4-5a, the distributions of the final performances for two experiments (fixed and annealed kernel sizes) are shown. In the static kernel case, the kernel size is kept fixed at $\sigma = 10^{-2}$, whereas the changing kernel had an exponentially annealed size $\sigma = 10^{-1} \to 10^{-2}$, during a training phase of 200 iterations. For this large static kernel size of $\sigma = 10^{-2}$, approximately 10% of the time the algorithm got trapped in a local maximum of the information potential with a normalized value of about 0.1. The algorithm avoided this local optimum in all the runs and achieved the global maximum, which has a normalized value of about 0.9, when the kernel size is annealed.

In Figure 4-5b, the distributions of the performances for three experiments are shown, but now the static kernel has a size of $\sigma = 10^{-3}$ throughout the training. The slow- and fast-annealed kernels, on the other hand, have exponentially decreasing sizes of $\sigma = 10^{-1} \to 10^{-3}$ for a training phase of 500 and 200 iterations, respectively. This annealing scheme is the same for all initial conditions. Since the kernel size is smaller now, we can expect more local maxima in the normalized information potential surface, but more accurate performance if global maximum is achieved (due to results in Table 4-1). In this small kernel case with $\sigma = 10^{-3}$, it is observed that the static kernel gets trapped in local maxima quite often (90% of the time), whereas, the fast annealed kernel shows some improvement in terms of avoiding local optima (70% of the time achieves global optimum), and eventually the slow annealed kernel consistently achieves the global maximum (100% of the time).



(a) (b)

Figure 4-5. Probability distributions of the final normalized information potential values obtained on the training set when the kernel size is a) large; static kernel (solid), slow annealing (+) b) small; static kernel (solid), fast annealing (+), slow annealing (dots)

These experiments showed that, by annealing the kernel size, one is likely to improve the algorithm's chances of avoiding local optimum solutions. However, there is no prescription for how to anneal the kernel size, yet. The exponential annealing scheme and the decay rates assumed in the above simulations were determined by trial and error.

4.3.2 Nonlinear System Identification Using a TDNN

The basic postulate behind the whole nonlinear dynamics research is that the dynamics of all deterministic nonlinear systems (without pure delays) can be represented as a first order multi-dimensional state equation (and perhaps stochastic systems also with some modifications). We also know from Kalman's work on the observability of linear dynamical systems and subsequent research on nonlinear systems that the state dynamics of any nonlinear set of equations can be replicated using an embedding of the output waveform [Tak81]. In fact, this is the idea behind the whole ARMA and nonlinear ARMA modeling approach. Consequently, TDNNs, which combine the embedding-in-time property of the delay-line with the universal approximation capabilities of MLPs [Bis95, Cyb89, Hor91], are a perfect match for this task. Therefore, as a second application, we investigate the performance of the minimum error entropy (MEE) criterion in identification of a nonlinear system using a TDNN.

Suppose we have samples from the input u_k and the output y_k of an unknown nonlinear system, where k is the discrete time index. The training set for the TDNN is constructed by embedding the input and the output sequences as follows.

$$\left\{ \begin{pmatrix} u_k & \dots & u_{k-L} & y_{k-1} & \dots & y_{k-M} \end{pmatrix}^T, y_k \right\} \qquad k = M, \dots, M+N-1 \tag{4.16}$$

In our simulations specifically, the embedding length of the input samples is chosen to be 7 (L = 6) and the embedding length of the output samples is chosen to be 6 (M=6). A two-layer TDNN with 7 hidden PEs is assumed upon suggestion [Pri99]. The *unknown* nonlinear system that is used to generate the input-output data has the following state dynamics and the output mapping.

$$
\begin{aligned}
x_{1,k+1} &= \left(\frac{x_{1,k}}{1+x_{1,k}^2} + 1 \right) \cdot \sin x_{2,k} \\
x_{2,k+1} &= x_{2,k} \cdot \cos x_{2,k} + \exp\left(-\frac{x_{1,k}^2 + x_{2,k}^2}{8} \right) + \frac{u_k^3}{1+u_k^2 + 0.5 \cdot \cos\left(x_{1,k} + x_{2,k}\right)} \\
y_k &= \frac{x_{1,k}}{1+0.5 \cdot \sin x_{2,k}} + \frac{x_{2,k}}{1+0.5 \cdot \sin x_{1,k}}
\end{aligned}
\tag{4.17}
$$

The training set consists of $N = 100$ input-output pairs as shown in Eq. (4.16) and the TDNN is trained starting from 50 randomly selected different initial conditions using both MEE (with quadratic entropy and Gaussian kernels) and MSE criteria. The output bias is, as usual, set to yield zero error mean over the training data. The performances of the optimal weights obtained from the two criteria are then compared on an independently generated 10000-sample test set.

Figure 4-6 shows the (zero-mean) error pdfs for the two criteria on this test set [Erd02c]. The MSE of the training error samples are 0.0676 and 0.0587, and the information potential for the same samples are 0.996 and 0.989 for MEE and MSE trained weights, respectively. As expected, the MSE is lower for the MSE-trained TDNN and information potential is higher for the entropy-trained TDNN in the training set.

This case study shows nicely the basic difference between entropy and variance minimization. Entropy prefers a larger and more concentrated peak centered at zero error with a number of small peaks at larger error values, whereas variance (MSE) prefers a wide-distributed error on a smaller range. In fact, this can be deduced by the following reasoning. Suppose it is possible to obtain many error distributions with the same variance. Since the Gaussian has the maximum entropy among fixed variance densities [Cov91], this error distribution would be the least desirable for the entropy criterion. Also the uniform would be among the non-desirable distributions for the error. The entropy would prefer rather spiky distributions, i.e., a number of δ-like concentrated spikes having the same variance. This is observed in Figure 4-6.

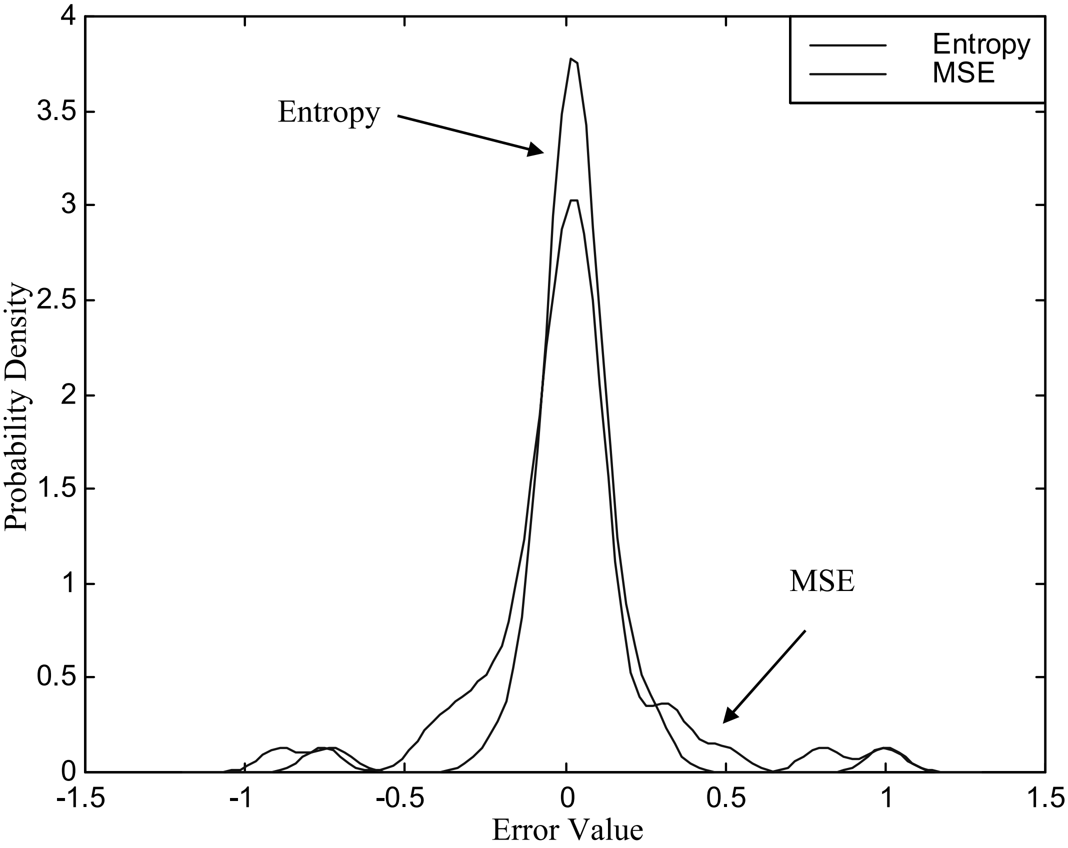


Figure 4-6. Probability density estimate of error values for MEE-trained (solid) and MSE-trained (dotted) TDNNs on the nonlinear system identification test data

A comparison of the desired output signal and the actual TDNN outputs using MEE-trained weights and MSE-trained weights is depicted in Figure 4-7 to show the improved statistical matching when MEE is used. Visual inspection of the pdf estimates of these three signals show that the MEE-trained TDNN approximates the pdf of the desired output much better around the most probable regions of the domain, when compared to the MSE-trained TDNN.

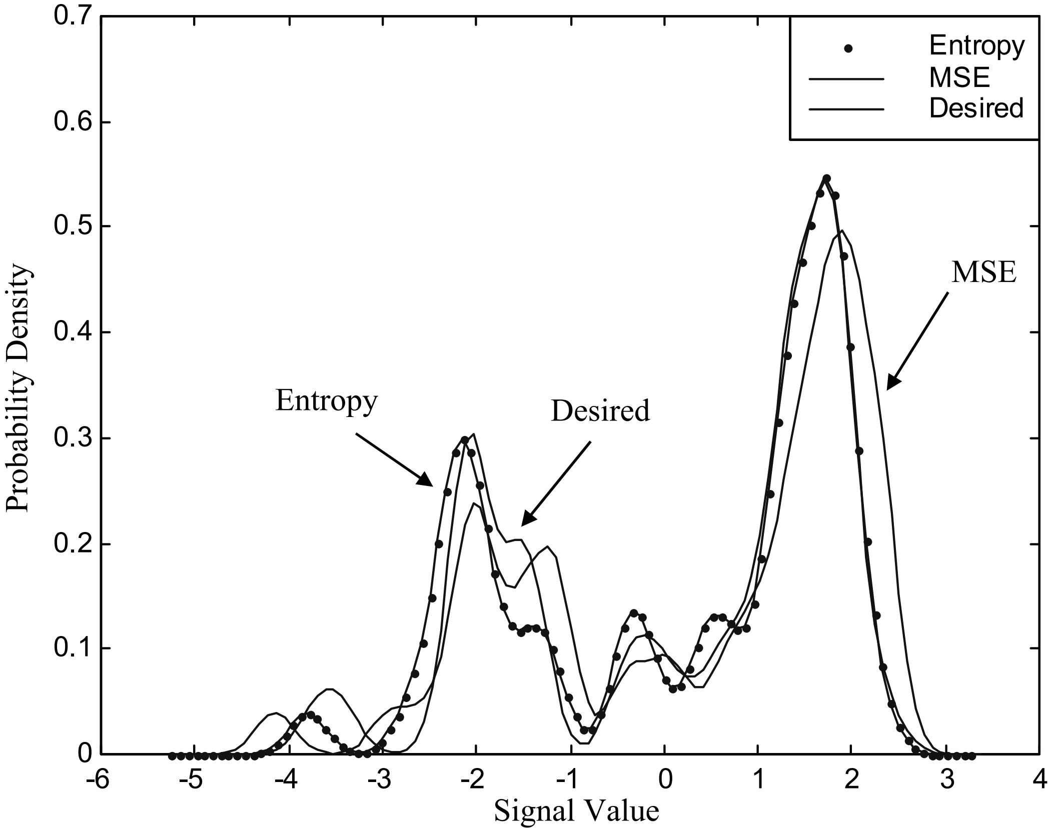


Figure 4-7. Probability density estimates for the desired (solid) and actual TDNN outputs using MEE-trained (thick dots) and MSE-trained (dotted) weights

4.3.3 Illustration of Global Optimization in the Generalized XOR Problem

This numerical case is designed to provide another experimental demonstration of the global optimization property of the entropy-training algorithm we propose. Namely, the 5-bit parity problem in the class of generalized XOR problems is considered. In this case study, a 5-5-1 MLP with *atan* nonlinearities in the hidden layer and a linear output PE is used. The 5 inputs take the values ± 1 according to the considered bit sequence and the desired output is again ± 1, specifically the XOR value corresponding to the input sequence, i.e., the one that makes the number of +1's even. The training set consists of all possible input sequences, numbering 32. In the static kernel case, the kernel size is set to $\sigma = 10^{-1}$ and the MLP is trained for 1000 iterations starting from 100 random initial weight vectors. In the annealed kernel case, the kernel size is annealed down exponentially as $\sigma = 10 \rightarrow 10^{-1}$ in 1000 iterations for the same initial conditions. It has been observed that the MLP trained using the annealed kernels achieved global optimum in all trials (100% of the time), whereas the MLP trained using the static kernels could rarely achieve the global optimum (10% of the time). The results of these experiments are summarized in Figure 4-8 [Erd02d].

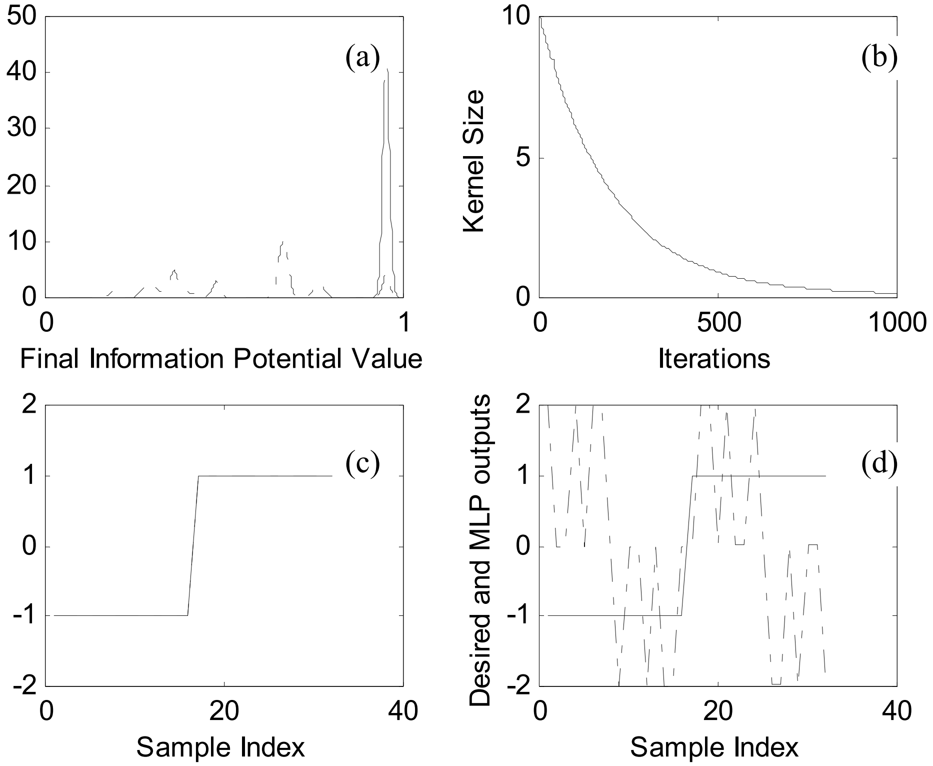


Figure 4-8. Results for the XOR problem a) Estimated probability densities of the final information potential values for the static kernel (dotted) and the annealed kernel (solid) cases b) Annealing of the kernel size versus iterations c) An illustration of the desired and the achieved output for the annealed kernel case, desired output (solid), MLP output (dotted) (due to perfect match not observed) d) An illustration of a local optimum from static kernel case, desired output (solid), MLP output (dotted)

In Figure 4-8a, the probability distribution of the final (normalized) information potential values is presented. Clearly, with the annealed kernels, the final information potential values are concentrated around the global maximum, whereas with the static kernels the algorithm is trapped at local maxima often. Figure 4-8b shows how the kernel size is exponentially annealed down in 1000 iterations. Figure 4-8c and Figure 4-8d show the MLP outputs with the optimal weights that exactly match the output to the desired and a local optimum that produces a low-grade output.

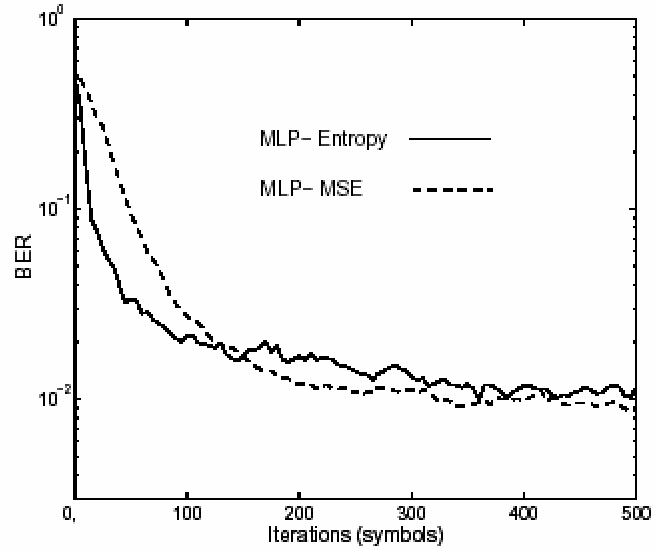


Figure 4-9. The BER versus iterations for MEE (solid) and MSE (dashed) criteria

4.3.4 Application to Nonlinear Channel Equalization for Digital Communications

We investigated the performance of the MEE criterion in channel equalization for digital communications [San02a]. As expected, we determined that under the linear channel model with additive Gaussian noise, the two criteria, i.e., MSE and MEE, provided identical solutions for the linear equalizer pointing out experimentally that under these circumstances using MSE and the associated LMS algorithm is more advantageous due to its relative computational simplicity. On the other hand, we obtained a faster convergence to small bit error rate (BER) values using MEE compared to MSE in nonlinear channel equalization with improvement in the distribution of the error samples at the output of the nonlinear equalizer, which is also a TDNN [San02a]. Using an online batch gradient descent algorithm that uses short sliding window of data to estimate the entropy and the associated gradient for weight updates, we train the TDNN to minimize the error entropy and the MSE (using LMS) in two different simulations. The results are as follows.

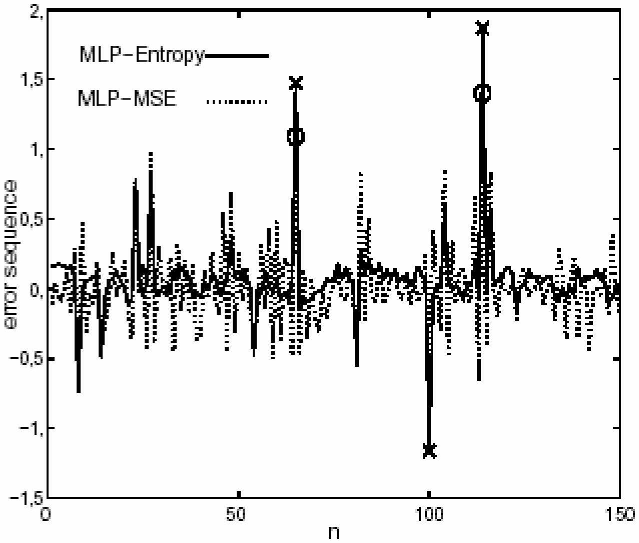


Figure 4-10. The error sequences at the output of the nonlinear equalizer for MEE (solid) and MSE (dotted) criteria; circles and crosses indicate the bit errors of MSE and MEE, respectively

The BER versus iterations plot shown in Figure 4-9 clearly show that the entropy trained equalizer is able to extract information from the data faster, i.e., using less samples, compared to MSE. Although the final value of the MSE is smaller in this case, the final performance of the MEE-trained network can be improved by perhaps increasing the length of the data window progressively as the iterations proceed. One other possibility to counter this problem of memory length falling short is to usilize a recursive estimator and a recursive gradient expression. This issue will be discussed in

Chapter 8, where a recursive gradient expression will be presented for use in gradient-based entropic learning scenarios.

The error sequences at the output of these two equalizers, shown in Figure 4-10 also provide valuable evidence to the basic difference between the preferences of MSE and MEE criteria. It can be observed in Figure 4-10 that while minimization of MSE results in a wide-spread error distribution, MEE prefers a more peaky concentration of samples around zero, while tolerating a few samples to be farther away. This is consistent with our expectations about the behavior of the algorithm and the results shown in Figure 4-6. Of course, this particular behavior is not desirable in terms of minimizing the BER. This is also evident from Figure 4-10; we observe that although the equalizer trained using entropy achieves a better fit to the inverse of the channel, it has more instances corresponding to wrong bit decisions compared with MSE. In that respect, MSE is more desirable than entropy. We also provide a fixed-point algorithm for fast adaptation of linear equalizers [San02a]. Although the applicability of this algorithm is limited to linear systems only, in the following subsection we will elaborate on an approximate algorithm to initialize MLPs using the least-squares principle, which could also be used as a sub-optimal training algorithm itself. In addition, the on-line gradient descent approach that has been used in the simulations can be replaced with a more principled *stochastic gradient* for entropy that is to be defined in the following chapters. This gradient algorithm will then be called the *stochastic information gradient*.

4.3.5 Demonstration of the Noise Rejection Capability of the MEE Criterion

In Theorem 4.2, we mentioned that the error entropy criterion is (ideally, i.e., when $N \to \infty$ and the parametric adaptive family of functions being trained span the target system being identified) robust to additive noise in the desired signal. This means, if we could obtain analytical expressions of the entropy values for each value of the weight vector and minimize these values, then the additive noise present in the desired signal would not be able to deviate the estimated system parameters from their corresponding actual values. In fact, MSE has the same noise rejection property asymptotically (which can be shown easily by demonstrating that under the conditions stated in the theorem, the error variance is minimized regardless of the noise variance when the adaptive system parameters match those of the unknown system).

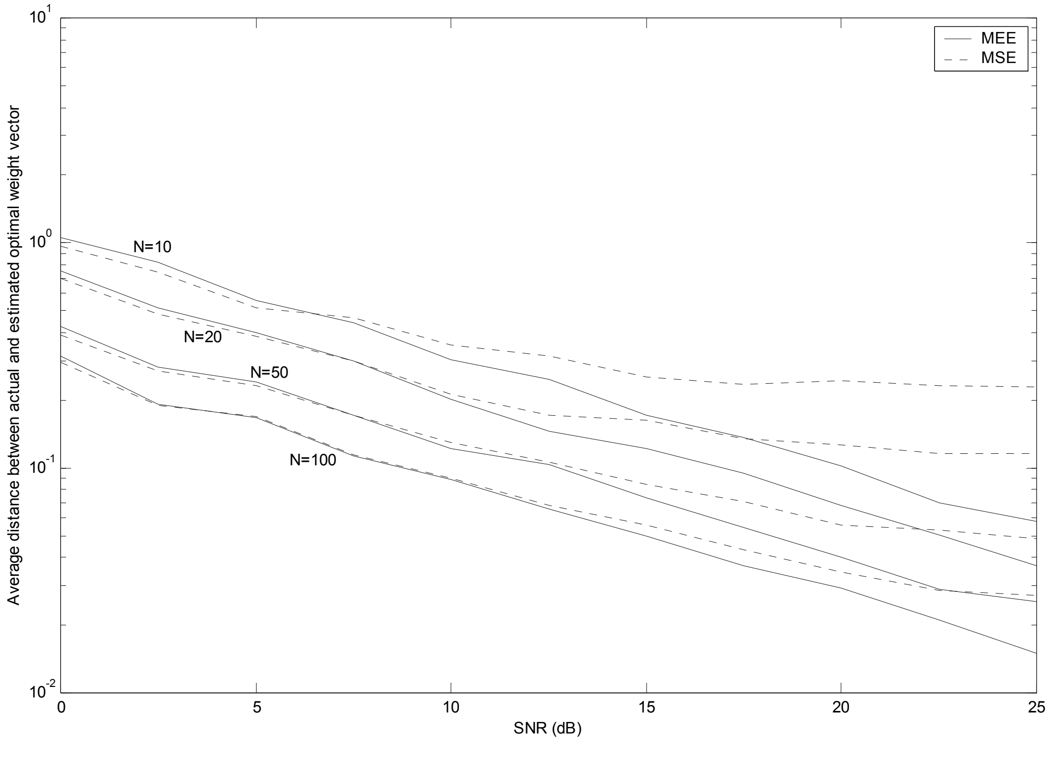


Figure 4-11. Average distance between the estimated and the actual weight vectors for MEE (solid) and MSE (dashed) criteria as a function of SNR using various sizes of training sets

In this section, we aim to show the noise rejection capability of the MEE criterion in finite-sample situations and compare this performance with that of MSE criteria. For simplicity, we will assume an ADALINE structure for the adaptive system with additive independent zero-mean Gaussian noise (this specific choice of the noise pdf has no significance) on the desired signal.

These experiments are repeated for various signal-to-noise-ratio (SNR) values in a Monte Carlo fashion. Specifically, for each SNR level, 100 Monte Carlo runs are performed using randomly selected training data. The adaptive system parameters are optimized through the use of gradient descent procedure for MEE (with $\alpha = 2$ and $\sigma = 1$ for the Gaussian kernels), and using the Wiener-Hopf equation for MSE (the covariance of the input and the cross-covariance of the desired signal and the input vector are estimated from the samples). For each run, the distance between the estimated weight vector for the model ADALINE and the actual weight vector of the *unknown* linear system (same structure as the adaptive system) is calculated. Then these are averaged over the 100 runs for each SNR value. Figure 4-11 shows the average deviation of the estimated weight vectors from the actual weight vectors for MEE and MSE as a function of SNR, using training set sizes of $N = 10,20,50,100$. Notice that for small noise power (SNR greater than approximately 5dB) MEE outperforms MSE in noise rejection consistently for all sizes of data sets. In addition, for high SNR values (greater than 20dB) MEE is extremely data efficient compared to MSE, because it obtains the same level of performance achieved by MSE using less samples. These results indicate that MEE is more robust to noise in the desired signal in a finite-sample case and furthermore it extracts the information efficiently to obtain a better solution with fewer samples.

An important issue in learning is the time required to obtain the optimal or an acceptable solution. When training using the MEE criterion (or any other criterion including MSE), in order to cut down the training time, it is desirable to initialize the adaptive system weights to a set of values that are close to the desired solution. There are many methods that are proposed to achieve the problem of *reducing the training time*. In Appendix D, we propose an efficient initialization scheme for MLPs, based on linear least squares. This approach can be used to accurately initialize the network weights to approximate the optimal solution and then the network training with MEE can proceed.

4.4 Structural Analysis of MEE Around the Optimal Solution for ADALINE

Suppose that the adaptive system under consideration in Figure 4-2 is an ADALINE structure with a weight vector *w*. The error samples are $e_k = d_k - w^T x_k$, where x_k is the input vector, formed by feeding the input signal to a tapped delay line for the special case of an FIR filter. When the entropy order is specified, minimizing the error entropy is equivalent to minimizing or maximizing the information potential for $\alpha < 1$ and $\alpha > 1$, respectively. Recall that the gradient of the information potential estimator with respect to the weight vector is simply

$$\frac{\partial V_\alpha}{\partial w} = \frac{(\alpha-1)}{N^\alpha} \sum_j \left(\sum_i \kappa_\sigma(e_j - e_i) \right)^{\alpha-2} \cdot \left(\sum_i \kappa'_\sigma(e_j - e_i)(x_i - x_j)^T \right) \tag{4.18}$$

In this expression, further simplifications are possible through the use of the scaling property of the kernel size and the following identity between the derivatives of a width-σ kernel and a unit-width kernel.

$$\kappa'_\sigma(x) = \frac{1}{\sigma^2} \kappa'(\frac{x}{\sigma}) \tag{4.19}$$

With these substitutions, the explicit expression for the gradient is easily determined to be as in Eq. (4.20).

$$\frac{\partial V_\alpha}{\partial w} = \frac{(\alpha-1)}{\sigma^\alpha N^\alpha}\sum_j\left(\sum_i \kappa\left(\Delta e_w^{ji}\right)\right)^{\alpha-2} \cdot \left(\sum_i \kappa'\left(\Delta e_w^{ji}\right)\cdot (x_i - x_j)^T\right) \tag{4.20}$$

From here on we will use the following notation.

$$\Delta e_w^{ji} = \left(\frac{\left(d_j - d_i\right) - w^T\left(x_j - x_i\right)}{\sigma}\right) \tag{4.21}$$

In order to maximize the information potential, we update the weights along the gradient direction with a certain step size η.

$$w(n+1) = w(n) + \eta \;\; \nabla V_\alpha(w(n)) \tag{4.22}$$

where $\nabla V_\alpha(w(n))$ denotes the gradient of V_α evaluated at $w(n)$.

To continue with our analysis, we consider the Taylor series expansion truncated to the linear term of the gradient around the optimal weight vector w_*.

$$\nabla V_\alpha(w) = \nabla V_\alpha(w_*) + \frac{\partial \nabla V_\alpha(w_*)}{\partial w}\left(w - w_*\right) \tag{4.23}$$

Notice that truncating the gradient at the linear term corresponds to approximating the cost function around the optimal point by a quadratic curve. The Hessian matrix of this quadratic performance surface is $R/2$, where R is given as

$$\begin{aligned} R = \frac{\partial \nabla V_\alpha(w_*)}{\partial w} = \frac{\partial^2 V_\alpha(w_*)}{\partial w^2} = \frac{(\alpha-1)}{\sigma^\alpha N^\alpha}\sum_j\left[\sum_i \kappa(\Delta e_{w_*}^{ji})\right]^{\alpha-3} \cdot \\ \left\{ \begin{array}{l} (\alpha-2)\left[\sum_i \kappa'(\Delta e_{w_*}^{ji})\cdot(x_i - x_j)\right]\cdot\left[\sum_i \kappa'(\Delta e_{w_*}^{ji})\cdot(x_i - x_j)^T\right] \\ +\left[\sum_i \kappa(\Delta e_{w_*}^{ji})\right]\cdot\left[\sum_i \kappa''(\Delta e_{w_*}^{ji})\cdot(x_i - x_j)(x_i - x_j)^T\right] \end{array} \right\} \end{aligned} \tag{4.24}$$

Now, defining a new weight vector space $\overline{w} = w - w_*$ whose origin is translated to the optimal solution w_*, we can rewrite the linearized dynamics of the weight equations in the vicinity of the solution in terms of the step size and the Hessian matrix as shown in Eq. (4.25). These are the coupled equations for the translated weights.

$$\overline{w}(n+1) = [I + \eta \ R]\overline{w}(n) \tag{4.25}$$

In order to obtain decoupled equations, we rotate the vector space by defining $v = Q^T \overline{w}$, Q being the orthonormal matrix consisting of the eigenvectors of R. Thus, the uncoupled dynamics for the translated and rotated weights are

$$v(n+1) = [I + \eta \ \Lambda]v(n) \tag{4.26}$$

where Λ is the diagonal eigenvalue matrix with entries ordered in correspondence with the ordering in Q. From this set of equations, we can isolate the dynamics of the weight vector along each mode of the matrix R. Specifically, for the i^{th} mode, the dynamic equation will only depend on the i^{th} eigenvalue of R by

$$v_i(n+1) = [1 + \eta \ \lambda_i]v_i(n), \qquad i = 1,\ldots,l \tag{4.27}$$

Note that, since R is the Hessian of the performance surface evaluated at a local maximum, its eigenvalues are negative. For a stable dynamics, all of the coefficients in the n equations of Eq. (4.27) must be inside the unit circle, that is $\left|1 + \eta \ \lambda_i\right| < 1$. This results in the following bound for the step size for stability.

$$0 < \eta \ < \frac{1}{\max_i \left|\lambda_i\right|} \tag{4.28}$$

This condition is similar to what we obtain for the MSE criterion [Far98, Hay84]; except, we consider the eigenvalues of the Hessian matrix of information potential instead of those of the covariance matrix (autocorrelation matrix in the FIR filter case) of the input vector to the ADALINE.

At this point, it also becomes possible to talk about time constants of the modes in the neighborhood of the optimum point. We can determine an approximate time constant for each individual mode whose dynamic equations are governed by Eq. (4.27). Specifically, for the k^{th} mode, we write

$$(1+\eta\ \lambda_k) = e^{-1/\tau_k} \tag{4.29}$$

from which the time constant is determined to be

$$\tau_k = \frac{-1}{\ln(1+\eta\lambda_k)} \approx \frac{-1}{\eta\lambda_k} = \frac{1}{\eta|\lambda_k|} \tag{4.30}$$

The time constants allow us to compare the convergence times of different modes. In order to evaluate the overall convergence speed, one must consider the slowest mode, which corresponds to the largest time constant, i.e., the largest (smallest in absolute value) eigenvalue. Understanding the relationship between the eigenvalues of the Hessian matrix in Eq. (4.24) and the two parameters, the kernel size and the entropy order, is crucial to maintain the stability of the algorithm following any changes in these parameters. One practical case where this relationship becomes important is when we adapt the kernel size during the training in connection with *Conjecture 4.1*. Since in this approach, the kernel size is decreased, we need to know how to adapt the step size to achieve faster learning in the initial phase of adaptation (by using a larger step size) and stable convergence in the final phase (by using a smaller step size). As an example, consider the case where we evaluate the quadratic information potential using Gaussian kernels. In this case, the Hessian matrix simplifies to

$$R = \frac{1}{\sigma^2 N^2}\sum_j\left[\sum_i \kappa''(\Delta e_{w_*}^{ji})(x_i - x_j)(x_i - x_j)^T\right] \tag{4.31}$$

Observe from Eq. (4.21) that as σ increases, $\Delta e_{w_*}^{ji} \to 0$, therefore, $\kappa''(\Delta e_{w_*}^{ji}) \to 0^-$ with speed $O\left(\sigma^{-6}\right)$. This is faster than the reduction rate of the denominator, which is $O\left(\sigma^{-2}\right)$, hence overall, the eigenvalues of R approach 0^-. This means that the valley near the global maximum gets wider and one can use a larger step size in steepest ascent, while still achieving stable convergence to the optimal solution. In fact, this result can be generalized to any kernel function and any α. The dilation effect mentioned in *Conjecture 4.1* is a direct cause of the increase in eigenvalues towards zero.

The analysis of the eigenvalues for varying α is more complicated. In fact, a precise analysis cannot be analytically pursued, but we can still try to predict as to how the eigenvalues of the Hessian behave as this parameter is modified. In order to estimate the behavior of the eigenvalues under changing α, we will exploit the following well-known result from linear algebra relating the eigenvalues of a matrix to its trace. For any matrix R, whose eigenvalues are given by the set $\{\lambda_i\}$, the following identity holds.

$$\sum_i \lambda_i = trace(R) \tag{4.32}$$

$$trace(R) = \frac{(\alpha-1)}{\sigma^{\alpha} N^{\alpha}} \sum_j \left[\sum_i \kappa(\Delta e_{w_*}^{ji}) \right]^{\alpha-3} \cdot \left\{ \begin{aligned} &(\alpha-2)\sum_k \left[\sum_i \kappa'(\Delta e_{w_*}^{ji}) \cdot (x_{ik} - x_{jk}) \right]^2 \\ &+ \left[\sum_i \kappa(\Delta e_{w_*}^{ji}) \right] \cdot \left[\sum_i \kappa''(\Delta e_{w_*}^{ji}) \cdot \left(\sum_k (x_{ik} - x_{jk})^2 \right) \right] \end{aligned} \right\} \tag{4.33}$$

Now consider the general expression of R given in Eq. (4.24). The trace of R is easily computed to be as given below in Eq. (4.33). The eigenvalues of R are negative and the dominant component, which introduces this negativity, is the term in the last line of Eq. (4.33). The negativity arises naturally since we use a differentiable symmetric kernel; since at w_* the entropy is small, the error samples are close to each other and the second derivative evaluates as a negative coefficient. Now let's focus on the term which involves the $(\alpha - 3)$-power in the first line of Eq. (4.33). Since all other terms vary linearly with α, this term will dominantly affect the behavior of the trace when α is varied. Consider the case where σ is small enough such that the small entropy causes the kernel evaluations in the brackets to be close to their maximum possible values and the sum therefore exceeds one. In that case, the power of the quantity in the brackets will increase exponentially with increasing α (for $\alpha > 3$), thus regardless of the terms affected linearly by α, the overall trace value will decrease (increase in absolute value). Consequently, a narrower valley towards the maximum will appear and the upper bound on the step size for stability will be reduced.

On the other hand, if the kernel size is large so that the sum in the brackets is less than one, then the $(\alpha - 3)$-power of this quantity will decrease, thus resulting in a wider valley towards the maximum in contrast to the previous case (for $\alpha > 3$). However, in practice we do not want to use a very small or a very large kernel size, as this will increase the variance or increase the bias of the Parzen estimation, respectively [Par67].

In fact, there is another approach that directly shows how the eigenvalues of R will decrease with increasing α and vice versa. Consider the expression in Eq. (4.24) or Eq. (4.33) again. Since at the operating point the error entropy is small and the difference between error samples is close to zero, the sums involving the derivative of the kernel function are approximately zero. Under the conditions mentioned in the previous paragraph, all the terms involving α remain as scalar coefficients that multiply a matrix, whose eigenvalues are negative. With the same arguments on how increasing α increases these coefficients, we conclude that the eigenvalues of the matrix R will increase in absolute value for a small kernel size and decrease for a large kernel size.

These conclusions are summarized in the following two facts.

Fact 4.1. Regardless of the entropy order, increasing the kernel size results in a wider valley around the optimal solution by decreasing the absolute values of the (negative) eigenvalues of the Hessian matrix of the information potential criterion.

Proof. In the preceding text.

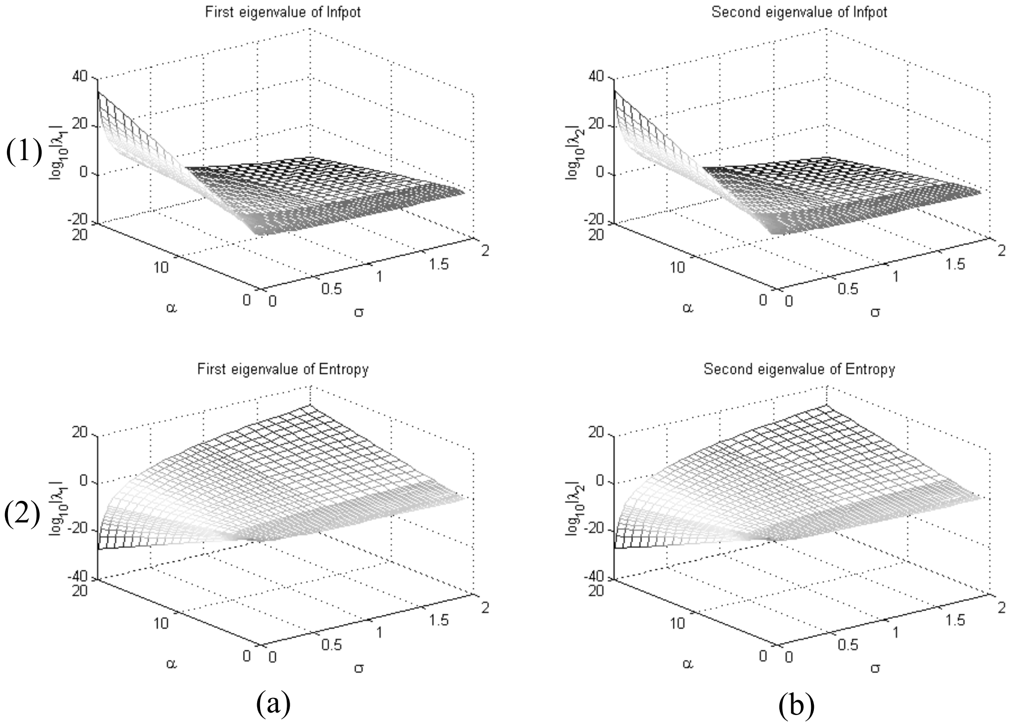


Figure 4-12. Log-absolute-value of the Hessian eigenvalues for the information potential (a1 and b1) and entropy (a2 and b2) evaluated at the optimal solution, presented as a function of the kernel size (σ) and the entropy order (α). There are two eigenvalues since the ADALINE has two weights

Fact 4.2. The effect of entropy order on the eigenvalues of the Hessian depends on the value of the kernel size. If the kernel size is small, then increasing the entropy order increases the absolute values of the (negative) eigenvalues of the Hessian of the information potential function at the global maximum. This results in a narrower valley. If the kernel size is large, the effect is the opposite, i.e., increasing the entropy order decreases the absolute value of the eigenvalues of the Hessian of the information potential, resulting in a wider valley. This analysis is expected to hold at least for $\alpha > 3$.

Proof. In the preceding text.

We remark that our conclusions in this section do not only apply to the eigenvalues of *R*, but they generalize to how these two parameters affect the volume of the region where our quadratic approximation is valid. These results are imperative from a practical point of view, because they explain how the structure of the performance surface can be manipulated by adjusting these parameters. Besides, they identify the procedures to adjust the step size for fast and stable convergence.

In order to show these results, we provide below a numerical case study, where we evaluate the eigenvalues of the Hessian of the information potential for a 2-weight ADALINE at its optimal weights for various values of the kernel size and entropy order in a supervised training scenario with a set of 20 noiseless training samples. The results are shown in Figure 4-12. In the subplots presented in the first row of Figure 4-12 it is clearly seen that the behavior of the absolute values of the eigenvalues of the information potential is exactly as we expected according to the theoretical analysis above. Note however, that the logarithms of the eigenvalues are pictured to account for a wider range of values and this is the reason why the values of the two eigenvalues look similar.

In the second row of Figure 4-12 we also presented the eigenvalues of the actual entropy evaluated at the optimal point, for the same values of kernel size and entropy order. Recall that entropy and information potential are related to each other by $H_\alpha(e) = [\log V_\alpha(e)]/(1-\alpha)$, hence their Hessians are associated to each other by

$$\frac{\partial^2 H_\alpha(e)}{\partial w^2} = \frac{1}{1-\alpha} \frac{V_\alpha(e) \cdot (\partial^2 V_\alpha(e) / \partial w^2) - (\partial V_\alpha(e) / \partial w) \cdot (\partial V_\alpha(e) / \partial w)^T}{V_\alpha^2(e)} \tag{4.34}$$

Since the error entropy is minimized (as opposed to the maximization of the information potential for $\alpha > 1$), the eigenvalues of its Hessian in Eq. (4.34) are positive already.

In this chapter, we introduced the principle of minimum error entropy training for information theoretic supervised learning of adaptive systems. We investigated the application of this principle to the training of nonlinear adaptive systems, specifically MLPs, in chaotic time-series prediction, nonlinear system identification and classification type problems. We showed the basic behavioral distinctions from the traditional MSE criterion. In order to cut down the training time in batch mode learning, we proposed an initialization algorithm that sets the weights of the MLP nearby the optimal solution.

CHAPTER 5
APPLICATION OF RENYI'S MUTUAL INFORMATION TO INDEPENDENT COMPONENT ANALYSIS

5.1 Brief Overview of Independent Component Analysis

Principal components analysis (PCA) is a tool that investigates the second-order statistical structure underlying a given random vector. Basically, the purpose of PCA is to identify a set of random variables that are uncorrelated with each other and maximize the variance [Oja83] that are linear projections of a random vector. Independent Component Analysis (ICA) is a generalization of this concept, in which the uncorrelatedness objective is further strengthened to independency of the projected random variables [Hyv01]. Besides its theoretical appeal, ICA is interesting in that it has found applications in signal processing problems under the name of instantaneous, linear blind source separation (BSS). The BSS problem seeks to separate a set of unknown source signals that are mixed by an unknown mixture. For the special case of an instantaneous and linear mixture, the ICA model can be applied to the problem although alternatives exist that exploit the time structure of the second-order statistical properties of the source signals [Dia01, Wu99]. In this section, we will only consider the ICA model for the BSS problem, which requires the conditions about the nature of the mixture mentioned above to hold (at least approximately).

In this respect, a typical BSS system consists of n observations that are linear combinations of m mutually independent source signals ($n \geq m$). Thus, the observation vector z, the source vector s, and the full column-rank mixing matrix H form the equation *z*=*Hs*. In the BSS literature, the square mixture case where the number of measurements is equal to the number of sources is the most investigated, because if there are more measurements than sources, for instance, PCA may be applied to select the *m* directions in the observation space that preserve most of the signal variability. This procedure may also be used to improve the SNR for the observations. In the BSS literature, in order to solve the square BSS problem, minimization of the mutual information (MMI) between outputs (estimated source signals) is considered to be the natural information theoretic criterion [Car98, Hyv99b, Yan97]. In spite of this understanding, two of the most well known methods for BSS, i.e., Bell and Sejnowski's InfoMax algorithm [Bel95], and Hyvarinen's FastICA [Hyv99a], use respectively the maximization of joint output entropy and fourth order cumulants (kurtosis). Shannon's mutual information can be written as the sum of Shannon's marginal entropies minus the joint entropy. One difficulty in using Shannon's MMI is the estimation of the marginal entropies. In order to estimate the marginal entropy, Comon and others approximate the output marginal pdfs with truncated polynomial expansions [Cho00, Com94, Hyv99b], which naturally introduces error in the estimation procedure. There are also parametric approaches to BSS, where the designer assumes a specific parametric model for the source distributions based on previous knowledge in the problem [Cho00]. A well-known result from statistical signal processing theory is that if the designer chooses an accurate parametric model for the problem, it will outperform any nonparametric approach, as the ones proposed in this chapter. However, it is also well known that the penalty for model mismatch is high, so there is an intrinsic compromise on the use of parametric modeling. An algorithm proposed by Xu *et al*. [Xu98] avoids the polynomial expansion by using the nonparametric Parzen windowing with Gaussian kernels to estimate directly Renyi's quadratic joint entropy at the output of the mapper as described in Chapter 2. Unfortunately, Xu's method requires estimation of the *n*-dimensional joint entropy, and non-parametric pdf estimation using Parzen windows looses robustness in high-dimensional spaces. The algorithm described in this chapter, avoids this shortcoming and has proved to be superior to many commonly accepted methods, because it requires less data to achieve the same performance level [Hil01a]. The algorithm presented by Hild *et al.* [Hil01a] was also restricted to Renyi's quadratic mutual information and Gaussian kernels in Parzen windowing using the old estimator for the marginal entropies. In this chapter, we will also use the generalized entropy estimator to investigate the effect of kernel and entropy order choice on the final separation performance.

It is necessary at this point to note that almost all of the simulation work on BSS presented in this thesis is performed by Kenneth E. Hild II, who is a colleague of mine at CNEL. The original idea of using the proposed cost function and the topology was his. Together, we proved that both the cost function and the topology are suitable to solve the problem. Kenneth prefers using the quadratic entropy estimator. With some modifications on his computer code, I completed the simulations that use other entropy orders and kernel functions. In any case, most of the credit for the work presented here on BSS should go to him.

5.2 Background for the BSS Algorithm

It is well known that an instantaneous, linear mixture can be separated by a spatial whitening (sphering) block followed by a pure rotation in *n* dimensions [Com94]. In fact, in general, for all BSS algorithms pre-whitening is suggested to increase convergence speed [Hyv01]. Our algorithm exploits this two-dimensional topology in solving the BSS/ICA problem with Renyi's mutual information.

In this approach, the spatial (pre-) whitening matrix is evaluated from the observed data; $W=Q\Lambda^{-1/2}$, where Q is the matrix of eigenvectors of the covariance matrix of the observations, and Λ is the corresponding eigenvalue matrix. Applying this transformation on the samples of the observation vector, $x=Wz$, we obtain the whitened samples x. The rotation matrix that follows this whitening procedure is adapted according to Renyi's mutual information to produce the outputs $y=R(\theta)x$. Here, θ denotes the set of Givens rotation angles that are used to parameterize the rotation matrix [Hil01a]. Now recall the following identity that holds for an n-dimensional random vector Y and its marginals, Y^o [Cov91], which relates the mutual information between the components of the random variable to the marginal entropies of these components and the joint entropy.

$$I_S(Y)=\sum_{o=1}^{n}H_S(Y^o)-H_S(Y) \tag{5.1}$$

The same equality is not valid for Renyi's definitions of these quantities because Renyi's entropy lacks this additivity property of Shannon's entropy. Nevertheless, we will show that we can slightly modify Renyi's mutual information expression such that it preserves the global minimum of the mutual information. Recall from Eq. (2.39) that Renyi's mutual information for an n-dimensional random variable Y is defined as

$$I_\alpha(Y)=\frac{1}{\alpha-1}\log\int_{-\infty}^{\infty}\frac{f_Y(y)^\alpha}{\prod_{o=1}^{n}f_{Y^o}(y^o)^{\alpha-1}}dy \tag{5.2}$$

However, the sum of Renyi's marginal entropies minus the joint entropy results in a ratio of integrals.

$$\sum_{o=1}^{n} H_\alpha(y^o) - H_\alpha(\mathrm{y}) = \frac{1}{\alpha - 1} \log \frac{\int_{-\infty}^{\infty} f_Y(\mathrm{y})^\alpha \, d\mathrm{y}}{\int_{-\infty}^{\infty} \prod_{o=1}^{n} f_{Y^o}(y^o)^\alpha \, dy} \tag{5.3}$$

Although this is not identical to Eq. (5.2), it is very similar in structure. In addition, Eq. (5.2) and Eq. (5.3) are both non-negative and they both evaluate to zero if and only if the joint pdf can be written as the product of the marginal densities, i.e., when the components of Y are mutually independent. This can be seen easily by letting the joint density, which is the integrand in the numerator, to be equal to the product of marginal densities, which is the integrand in the denominator. In that case, the argument of the logarithm in Eq. (5.3) becomes unity, hence the minimum value of zero is achieved, and thus the sources are separated. On the other hand, if the right hand side in Eq. (5.3) becomes zero, the argument of the logarithm becomes unity, thus the numerator is equal to the denominator. For this to occur between a joint distribution and its marginals, it is necessary for the marginal random variables to be independent. Having proved that the expression in Eq. (5.3) is a valid criterion for measuring independence (that is its global minimum occurs when its arguments are independent), we adopt it as the cost function instead of the actual mutual information, given in Eq. (5.2).

Since only the rotation matrix in the separating topology is adapted to minimize Eq. (5.3) and since Renyi's joint entropy is invariant to rotations (see Property 2.5), we can remove this term and reduce the cost function to Eq. (5.4) which mimics the cost functions of Comon and Yang [Com94, Yan97] with Renyi's entropy substituted for Shannon's. In order to estimate the marginal entropies of each output Y^o, we will use our nonparametric estimator.

$$J(\theta) = \sum_{o=1}^{n} H_\alpha(Y^o) \tag{5.4}$$

The Givens rotation parameter vector θ consists of $n(n\text{-}1)/2$ parameters θ_{ij}, $j>i$, where each parameter represents the amount of Givens rotation in the corresponding *i-j* plane. The overall rotation matrix is the product of the individual in-plane rotation matrices

$$R(\theta) = \prod_{i=1}^{n-1} \prod_{j=i+1}^{n} R_{ij}(\theta_{ij}) \tag{5.5}$$

In Eq. (5.5), all products are performed sequentially from the right (or left). The important point is to perform these operations in the same order and from the same side when evaluating the gradient expression. The Givens rotation in the *i-j* plane is defined as an identity matrix whose $(i,i)^{\text{th}}$, $(i,j)^{\text{th}}$, $(j,i)^{\text{th}}$, and $(j,j)^{\text{th}}$ entries, as in a rotation in two-dimensions, are modified to read $\cos\theta_{ij}$, $-\sin\theta_{ij}$, $\sin\theta_{ij}$, and $\cos\theta_{ij}$, respectively.

5.3 Adaptation Algorithm for the Rotation Matrix

The batch mode adaptation algorithm for the rotation matrix, which is parameterized in terms of Givens rotations, can be summarized as follows.

Algorithm 5.1. Batch mode BSS algorithm using Renyi's entropy.

1. Whiten all observation samples $\{z_1,...,z_N\}$ using W to produce the samples $\{x_1,...,x_N\}$.
2. Initialize (randomly) the Givens rotation angles θ_{ij}, $i = 1,...,n-1$, $j = i+1,...,n$.
3. Compute the rotation matrix using Eq. (5.5) and evaluate the output samples using it.
4. Until the algorithm converges repeat the following steepest descent procedure
   a. Evaluate the gradient of the cost function $J(\theta) = \sum_{o=1}^{n} \hat{H}_\alpha(Y^o)$, using

$$\frac{\partial J}{\partial \theta_{ij}} = \sum_{o=1}^{n} \frac{\partial \hat{H}_{\alpha}(Y^{o})}{\partial \theta_{ij}} = \sum_{o=1}^{n} \frac{1}{1-\alpha} \frac{\partial \hat{V}_{\alpha}(Y^{o}) / \partial \theta_{ij}}{\hat{V}_{\alpha}(Y^{o})} \tag{5.6}$$

$$y_j^o = R^o x_j \quad o = 1,\ldots,n, \quad j = 1,\ldots,N$$

$$\frac{\partial y_j^o}{\partial \theta_{ij}} = \frac{\partial R^o}{\partial \theta_{ij}} x_j = \left(\frac{\partial R}{\partial \theta_{ij}} \right)^{o} x_j \tag{5.7}$$

$$\frac{\partial R}{\partial \theta_{ij}} = \left(\prod_{p=1}^{i-1} \prod_{q=p+1}^{n} R_{pq} \right) \left(\prod_{q=i}^{j-1} R_{iq} \right) R'_{ij} \left(\prod_{q=j+1}^{n} R_{iq} \right) \left(\prod_{p=i+1}^{n} \prod_{q=p+1}^{n} R_{pq} \right) \tag{5.8}$$

where for any matrix A, A^o denotes the o^{th} row of that matrix and R'_{ij} denotes the derivative of the specific Givens rotation matrix (in the *i-j* plane) with respect to its parameter θ_{ij}.

b. Update the Givens angles using

$$\theta_{ij} \leftarrow \theta_{ij} - \eta \frac{\partial J}{\partial \theta_{ij}} \tag{5.9}$$

The algorithm above is for the separation of real-valued signals from real-valued mixtures. In order to generalize it to the case of complex-valued mixtures, the Givens matrices must be modified by incorporating imaginary parts to the rotation angles to account for rotations in the imaginary portions of the complex-valued vector space.

5.4 Simulations for Batch Mode BSS Using Renyi's Entropy

The whitening-rotation scheme has a very significant advantage. We observed experimentally that when this topology is used, with a large number of samples, there are no local minima of the cost function. Consider a 2-source separation problem, for instance. The rotation matrix consists of a single parameter, which can assume values in the interval [0,2π). As far as separation is concerned, there are four equivalent solutions, which correspond to two permutations of the sources and the two possible signs for each source. The value of the cost function is periodic with $\pi/2$ over the rotation angle θ, and most often is a very smooth function (sinusoidal like), which is easy to search using descent based numerical optimization techniques.

Numerous simulations were performed with this new BSS algorithm using different α and smooth kernels on synthetic and audio data instantaneous mixtures. In order to compare the results from different algorithms, we used a signal-to-distortion ratio (SDR), which is defined as

$$SDR = \frac{1}{n}\sum_{i=1}^{n} 10\log_{10}\left(\frac{(\max q_i)^2}{q_i q_i^T - (\max q_i)^2}\right) \tag{5.10}$$

where $q=RWH$ is the overall mixing+separation matrix and q_i is the i^{th} row of q. This criterion effectively measures the distance of q from an identity matrix and is invariant to permutations and scaling.

We first start with an investigation of the effect of α on the separation of instantaneous mixtures, when the source kurtosis values span the range of super- and sub-Gaussian signals. Although our nonparametric method in principle can separate signals independent of their kurtoses (since the pdf is estimated at the output directly), a question of paramount importance is ‘what value of entropy order should one use for different source densities in order to achieve optimal performance?’ supposing that we have some knowledge on this aspect of the problem. In search of the answer to this question, a series of Monte Carlo simulations are performed (10 for each), using source distributions of different kurtosis values. In all these simulations, the two sources are assumed to have the same generalized Gaussian density, which is given by $G_\upsilon(x) = C \cdot \exp(-|x|^\upsilon / (\upsilon E[|x|^\upsilon]))$.

The parameter υ controls the kurtosis of the density and this family includes distributions ranging from Laplacian $(\upsilon = 1)$ to uniform $(\upsilon \to \infty)$. Gaussian distribution is a special case corresponding to $(\upsilon = 2)$, which leads to the classification of densities as super-Gaussian and sub-Gaussian for $(\upsilon < 2)$ and $(\upsilon > 2)$, respectively. For a given kurtosis value the training data set is generated from the corresponding generalized Gaussian density and a random mixing matrix is selected. Then the separation is performed using various entropy orders (tracing the interval from 1.2 to 8 in steps of 0.4) and Gaussian kernels. The Gaussian kernel size was set at 0.25, and the adaptation using *Algorithm 5.1* was run until a convergence of the SDR within a 0.1dB band was achieved (although in practice this cannot be used as the stopping criterion), which usually occurred in less than 50 iterations with a step size of 0.2.

Table 5-1. Optimal entropy order versus source density kurtosis

Kurtosis of sources		Optimal entropy order
Super-Gaussian Sources	0.8 $(\upsilon = 1)$	6.4
	0.5 $(\upsilon = 1.2)$	5.2
	0.2 $(\upsilon = 1.5)$	2
Sub-Gaussian Sources	-0.8 $(\upsilon = 4)$	1.2
	-0.9 $(\upsilon = 5)$	1.6
	-1.0 $(\upsilon = 6)$	1.2

According to these simulations, the optimal entropy orders for the corresponding kurtosis value of the source densities are determined and are presented in Table 5-1. These results indicate that, for super-Gaussian sources, entropy orders greater than or equal to 2 should be preferred, whereas for sub-Gaussian sources, entropy orders smaller than 2, perhaps closer to 1 or even smaller than 1, should be preferred. These results are in conformity with our expectations from the analysis of the information forces in Chapter 3. As we saw in that analysis, entropy orders larger than 2 emphasize samples in concentrated regions of data, whereas smaller orders emphasize the samples in sparse regions of data. If the mixtures belong to different kurtosis classes, then the quadratic entropy can be used as it puts equal emphasis on all data points regardless of their probability density. This effect is very different from some of the available algorithms where the BSS algorithms diverge if the kurtoses of the sources are misestimated [Bel95, Hyv99a, Hyv99b]. Another interesting aspect of this simulation is that it seems to imply that Shannon information definition ($\alpha \to 1$) is not particularly appropriate for separating super-Gaussian sources, although it might be useful for sub-Gaussian sources [Erd02b].

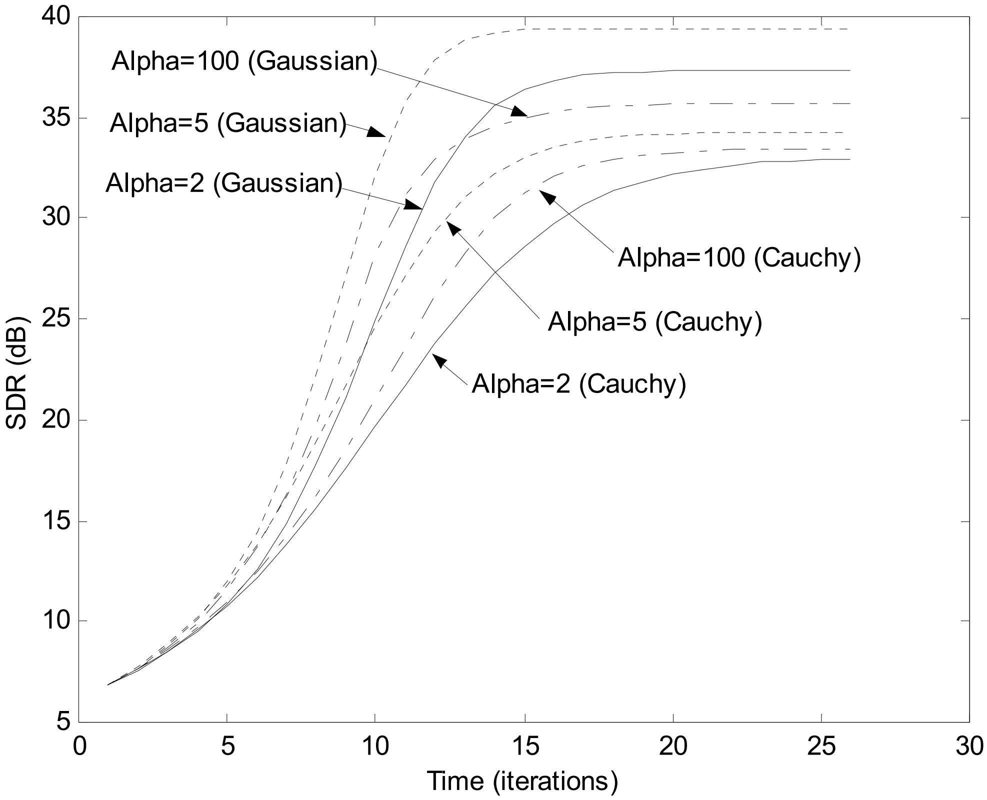


Figure 5-1. Evolution of the signal-to-distortion ratio during the iterations for two sources for different choices of entropy order and kernel function

The next question addresses the performance of *Algorithm 5.1* for a realistic source such as speech. Figure 5-1 shows the evolution of the SDR values as a function of number of iterations in a 2-audio-source problem for different choices of the kernel function and the parameter α. These plots clearly show that for both kernels, a better separation is achieved when α=5 is used. Also, we observe that the solutions generated using the Gaussian kernel are better than those generated with the Cauchy kernel, which hints at the possibility of determining an optimal kernel choice for a given data set. There is no significant deterioration of performance when α changes from 5 to 2, since both provide SDRs larger than 30 dB (20 dB, corresponding to a 100-to-1 SDR, is considered an acceptable separation performance). We also see that for very large α values performance deteriorates due to the smoothing effect in the kernel.

For a final comparison, we show the SDR plots in Figure 5-2 for our BSS algorithm with α =2 and Gaussian kernels, the FastICA (FICA) [Hyv99a] with the symmetric approach and the cubic nonlinearity, Infomax [Bel95] with Amari's natural gradient [Ama96], and Comon's minimization of mutual information (MMI) using an instantaneous mixture of 10 audio sources [Erd02b]. The sources consist of one music piece, four female and five male speakers. Spatial pre-whitening is used for each method, and the mixing matrix entries were chosen from a uniform density on [-1,1]. The numbers in parentheses are the number of data samples used to train each algorithm. It is clearly seen from the figure, that the MRMI method achieves better performance, although it uses a smaller data set. The improved data efficiency of the MRMI method is discussed and showed in greater detail by Hild *et al.* [Hil01a]. We attribute it to the fact that our method directly estimates the entropy and captures the information in the samples.

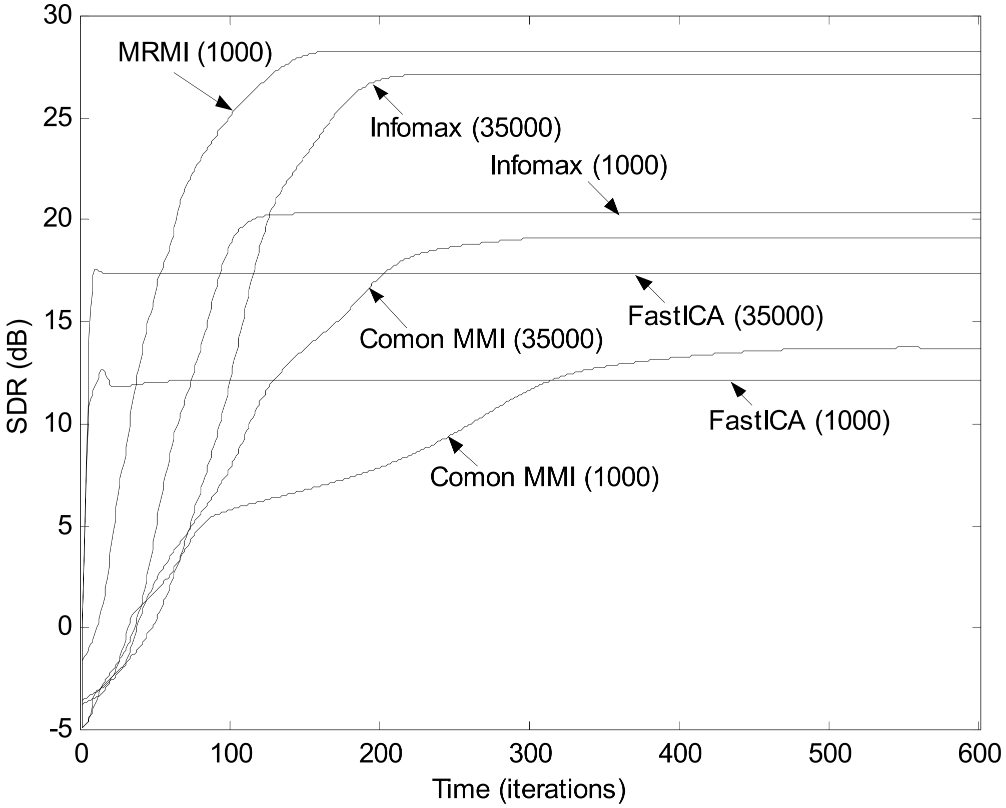


Figure 5-2. SDR versus iterations for our algorithm (MRMI), Infomax, FICA, Comon's MMI using the designated number of samples for the separation of 10 audio sources

Although the MRMI method converges in fewer iterations than the others do, keep in mind that it has $O(N^2)$ computational complexity per update as compared to $O(N)$ for the other two methods. Yang and Amari's MMI algorithm was also applied to this problem, however, we were never able to achieve an acceptable separation level; therefore the corresponding results are not included in the figure.

CHAPTER 6
APPLICATION OF RENYI'S ENTROPY TO BLIND DECONVOLTION AND BLIND EQUALIZATION

6.1 Brief Overview of Blind Deconvolution and Blind Equalization

For the sake of simplicity, we consider here the discrete-time processes. Suppose we have the result of the convolution of two discrete-time sequences s_n and h_n, and let this new sequence be denoted by x_n, i.e., $x_n=h_n*s_n$. In a realistic context, these two sequences could be the input signal to a linear time-invariant (LTI) channel and the impulse response of that channel. Suppose that we know neither the channel's impulse response nor the input sequence; except we could have some knowledge on the statistical properties of the input sequence and we have the output measurements x_n. In the classical literature, the problem of determining the input sequence only using this given information (up to an uncertainty in the sign, amplitude and the delay) is referred to as *blind deconvolution* and the problem of determining the channel impulse response's inverse is referred to as *blind equalization* [Hay94, Hay00b]. In time, as the interest of researchers shifted towards the application of this methodology to the blind equalization of digital communication channels, this term started specifically referring to digital blind equalization [Hay94, Hay00b, Nik93] and blind deconvolution fused with blind source separation (convolutive mixtures) and started being referred to as multi-channel blind deconvolution [Chi01, Jan00]. It is also known that, in essence, the two problems, blind deconvolution and blind equalization, are equivalent since convolution is a commutative operator [Hay94, Hay00b].

Throughout this dissertation, influenced by the recent nomenclature in the blind adaptation techniques literature, we will use the name *blind deconvolution* (BD) to describe the problem of determining the unknown input signal to an unknown channel and we will refer to the process of equalizing a digital communication channel as *blind equalization* (BE).

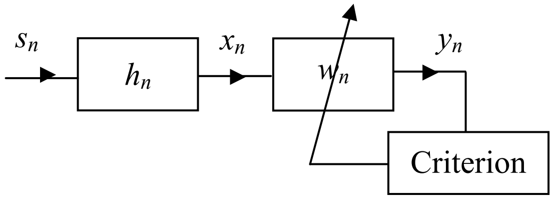


Figure 6-1. Typical blind deconvolution/equalization scheme with an appropriate criterion. The channel impulse response h_n and the input signal s_n are unknown

Typically, the blind deconvolution problem is represented by the block diagram given in Figure 6-1. Usually the equalizer w_n is parameterized as an FIR filter and a suitable criterion is determined depending on the assumptions on the statistical behavior of the input signal. Although in reality the measurement samples, x_n, may be corrupted by additive noise, we do not consider that case in our simulations. However, we will briefly mention the theoretical expectations of the proposed algorithms' robustness to noise.

There are some technical issues in the choice of the equalizer structure for various possibilities of the pole-zero locations of the *unknown* channel. First of all, notice that if the channel is a minimum/maximum-phase filter, then knowledge of the power spectral density (PSD) of the input signal is sufficient to determine the unique solution to the blind deconvolution problem; just evaluate the PSD of the measured x_n and divide it by the PSD of the input signal to get an estimate of the magnitude-squared frequency response of the channel. Then using the knowledge that it is minimum- or maximum-phase, one can determine easily the phase response also, thus completing the frequency domain representation of the channel [Hay94]. More important are the restrictions that the structure of the channel filter imposes on the structure of the equalizer. Explicitly, if the channel is minimum-phase, then a stable and causal inverse exists. If the channel is maximum-phase, then a stable and anti-causal inverse exists. If the channel is neither minimum- nor maximum-phase, then the stable inverse is non-causal (non-anti-causal also). Since, in general, it is not possible to know the relative locations of the zeros and the poles of the unknown channel filter with respect to the unit circle, the designer is forced to assume the worst case and use a non-causal FIR (in both directions) equalizer filter structure. The realistic requirement that the solution must be obtained in finite time (perhaps pre-specified) also limits the extent of the FIR filter in both the advance and delay directions. Typically the equalizer filter is chosen an FIR with tap weights $w_{-L},\ldots,w_L$; i.e., with equal length in both directions, although not necessary. This structural issue, however, seems to over-occupy most of the researchers proposing blind deconvolution algorithms [Ben80, Hay94]. In reality, the choice of the criterion has nothing to do with this structural issue. Besides, the performance of the algorithm does not vary much once the FIR filter length L is chosen sufficienty large such that the causal and anti-causal portions of the equalizer impulse response can accurately approximate the ideal stable inverse filter (i.e., has most of the energy in the available taps).

As for the choice of the criterion, many possibilities exist depending on the assumptions on the input signal's statistical structure. Donoho summarized the well-known approach of entropy minimization to solve this problem [Don81]. Minimum entropy deconvolution assumes that the source signal is a non-Gaussian distributed wide-sense-stationary (WSS), white process. Since at the time, effective entropy estimators were not available, the methods summarized by Donoho and contemporaries usually adopted higher order moments, which mimic the properties of entropy, of the signals under investigation, e.g. the kurtosis. In some cases, the designer may have knowledge about some certain statistics of the input signal and this information may be used to obtain better deconvolution results. For example, if the source probability density function (pdf) is known and if the source signal samples are assumed to be *iid*, then the maximum entropy approach may be used [Bel95].

To summarize, there are mainly two approaches that make use of the entropy as the adaptation criterion: minimum entropy deconvolution and maximum entropy deconvolution. Both approaches assume that the samples s_n are *iid* and the equalizer structure is chosen such that it can successfully (even if approximately) invert the channel filter. In the former approach, the topology shown in Figure 6.1 is used where the criterion is the entropy of the output of the equalizer. Due to the Benveniste-Goursat-Ruget theorem [Ben80], the entropy of the output of the equalizer is minimized if and only if the overall filter, h_n*w_n, is an impulse (with arbitrary delay and scale). The main intuition behind this approach is that, as the overall filter departs from an impulse, the output distribution approaches a Gaussian density. It is known that under fixed variance, Gaussian density has the maximum entropy and thus entropy can be used as a Gaussianity measure, i.e., minimizing entropy under the fixed variance constraint maximizes non-Gaussianity. In fact, basic ICA algorithms like our MRMI in Chapter 5, Comon's MMI and Hyvarinen's FastICA can also be regarded in this context as maximizing the average non-Gaussianity of the outputs by minimizing the sum of output marginal entropies or kurtoses [Hyv01]. In the latter approach (maximum entropy deconvolution), it is necessary to have an accurate estimate of the pdf of the source signal. From this pdf, the cdf of the source is determined and this function is introduced in Figure 6.1 as a nonlinear mapping on the output of the equalizer. As described by Bell and Sejnowski [Bel95], when the entropy after this nonlinearity is maximized, the distribution approaches to a uniform density on the interval [0,1]. This forces the pdf of the signal before the nonlinearity to approach that of the source signal and once again due to the Benveniste-Goursat-Ruget theorem the overall filter approaches an impulse function.

The existence of the nonlinearity limits the number of solutions to two (if the source pdf is even symmetric) for each possible delay amount, in the maximum entropy deconvolution scheme, corresponding to opposite sign weight vectors for the equalizer. In the minimum entropy scheme, however, there are infinitely many solutions for the optimal weight vector that lie on a line passing through the origin. Different solutions on the line correspond to different positive and negative scaling factors of the unit norm solution. Since the scale factor is an indeterminacy in the blind deconvolution problem, this does not pose any trouble, except when one tries to impose the constant variance constraint by constraining the weight vector to have unit norm at all times. There are two ways to achieve this. The commonly used approach in such situations is to normalize the weight vector after each weight update (see Oja’s rule for PCA for example) [Oja83, Hay99]. The second approach is to express the weight vector in spherical coordinates and parameterize the $(2L+1)$-length weight vector in terms of $2L$ directional angles and a unit norm. For example, in the 3-tap equalizer case, the unit norm weight vector would be written in terms of the two directional angles as $[\cos\theta_1 \cos\theta_2 \quad \sin\theta_1 \cos\theta_2 \quad \sin\theta_2]^T$. The adaptation would then be carried over these angles; however, this will introduce trigonometric evaluation requirements on the algorithm, which may not always be desirable.

A second approach in the minimum entropy deconvolution approach is to use a scale invariant cost function, so that the performance evaluations of two weight vectors that are co-linear but different in norm and sign yield the same value, thus not prefer one over the other. In the following sections, we will give an example of such an entropy-based scale-invariant cost function and use it in training an equalizer. In the following sections of this chapter, we will only consider the minimum entropy approach, however, we emphasize that the estimator we proposed for Renyi's entropy can easily and successfully be applied to the blind deconvolution problem in the maximum entropy sense as we described above.

6.2 Minimum Entropy Deconvolution Using Renyi's Entropy

In this section, we will provide a motivation for using Renyi's entropy as a criterion for minimum entropy blind deconvolution. As an initial step, consider the following theorem, which gives the relationship between the entropies of linearly combined random variables.

Theorem 6.1. Let S_1 and S_2 be two independent random variables with pdfs $p_{S_1}(.)$ and $p_{S_2}(.)$, respectively. Let $H_\alpha(.)$ denote the order-α Renyi's entropy for a continuous random variable. If a_1 and a_2 are two real coefficients in $Y=a_1S_1+a_2S_2$, then

$$H_\alpha(Y) \geq H_\alpha(S_i) + \log|a_i|, \quad i=1,2 \tag{6.1}$$

Proof. Since S_1 and S_2 are independent, the pdf of Y is given by

$$p_Y(y) = \frac{1}{|a_1|} p_{S_1}(y/a_1) * \frac{1}{|a_2|} p_{S_2}(y/a_2) \tag{6.2}$$

Recall the definition of Renyi's entropy for Y and consider the following identity.

$$\begin{aligned} e^{(1-\alpha)H_\alpha(Y)} &= \int_{-\infty}^{\infty} p_Y^\alpha(y)dy \\ &= \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} \frac{1}{|a_1 a_2|} p_{S_1}(\frac{\tau}{a_1}) p_{S_2}(\frac{y-\tau}{a_2}) d\tau\right] dy \end{aligned} \tag{6.3}$$

Using Jensen's inequality for convex and concave cases, we get

$$\begin{aligned} e^{(1-\alpha)H_\alpha(Y)} &\overset{\alpha>1}{\underset{\alpha<1}{\overset{\le}{\ge}}} \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} \frac{1}{|a_1|} p_{S_1}(\frac{\tau}{a_1})\left[\frac{1}{|a_2|} p_{S_2}(\frac{y-\tau}{a_2})\right]^\alpha d\tau\right] dy \\ &= \int_{-\infty}^{\infty} \frac{1}{|a_1|} p_{S_1}(\frac{\tau}{a_1})\left[\int_{-\infty}^{\infty}\left[\frac{1}{|a_2|} p_{S_2}(\frac{y-\tau}{a_2})\right]^\alpha dy\right] d\tau \\ &= \int_{-\infty}^{\infty} \frac{1}{|a_1|} p_{S_1}(\frac{\tau}{a_1}) V_\alpha(a_2 S_2) d\tau \\ &= V_\alpha(a_2 S_2) \cdot \int_{-\infty}^{\infty} \frac{1}{|a_1|} p_{S_1}(\frac{\tau}{a_1}) d\tau = V_\alpha(a_2 S_2) \end{aligned} \tag{6.4}$$

Reorganizing the terms in the last inequality and using the relationship between entropy and information potential, regardless of the value of α and the direction of the inequality, we arrive at the conclusion $H_\alpha(Y) \ge H_\alpha(S_i) + \log|a_i|$, $i=1,2$.

An immediate extension of this theorem is obtained by increasing the number of random variables in the linear combination to n.

Corollary 6.1. If $Y=a_1S_1+\ldots+a_nS_n$, with *iid* $S_i \sim p_S(.)$, then the following inequality holds for the entropies of S and Y.

$$H_\alpha(Y) \ge H_\alpha(S) + \frac{1}{n}\log|a_1 \ldots a_n| \tag{6.5}$$

where equality the two entropies occur iff $a_i = \delta_{ij}$, where δ denotes the Kronecker-delta function.

Proof. It is trivial to generalize the result in Theorem 6.1 to n random variables using mathematical induction. Thus, for the case where all n random variables are identically distributed we get n inequalities.

$$\begin{aligned} H_\alpha(Y) &\geq H_\alpha(S) + \log|a_1| \\ H_\alpha(Y) &\geq H_\alpha(S) + \log|a_2| \\ &\vdots \\ H_\alpha(Y) &\geq H_\alpha(S) + \log|a_n| \end{aligned} \tag{6.6}$$

Adding these inequalities, we get the desired result. The necessary and sufficient condition for the equality of entropies is obvious from the formulation. If $a_i = \delta_{ij}$, then Y=S, therefore the entropies are equal. If $a_i \neq \delta_{ij}$, then due to Theorem 6.1, entropy of Y is greater than the entropy of S (assuming normalized coefficients). Notice that this corollary does not necessarily guarantee that the entropy of Y is lgreater than the entropy of S, because when the absolute value of the product of the gains is less than one, the logarithm brings in a negative component. Through complicated algebra, which we omit here, we were able to show that in the vicinity of a combination where only one of the gains is close to one and all the others are close to zero, these terms lose their significance and the inequality is mostly dominated by the two entropy terms.

Notice that the blind deconvolution problem is structurally very similar to the situation presented in the above corollary. In that context, the coefficients a_i of the linear combination are replaced by the impulse response coefficients of the overall filter h_n*w_n. In addition, the random variables S and Y are replaced by the source signal and the deconvolving filter (equalizer) output signal, respectively. Especially, when close to the ideal solution, i.e., when h_n*w_n is close to an impulse, the second term in *Corollary 6.1* will approach rapidly to zero and the two entropy values will converge as the two signals *Y* and *S* converge to each other.

We mentioned in the previous section that in the minimum entropy approach to blind deconvolution, there are two possible methods to avoid the adaptation being dominated by the norm of the weight vector (i.e., equalizer tap weights). We will take here the second approach where the weight vector is not constrained but the cost function is made scale invariant.

The entropy of a random variable is not scale invariant. In order to solve the blind deconvolution problem using unconstrained optimization techniques and without having to normalize the weights at the end of every iteration, we need to modify this cost function by introducing appropriate terms to make it scale invariant. For this purpose, consider the following modified cost function.

Fact 6.1. The modified cost function

$$J(Y) = H_\alpha(Y) - \frac{1}{2}\log[Var(Y)] \tag{6.7}$$

is scale invariant. That is $J(aY) = J(Y)$, $\forall a \in \Re$.

Proof. It is trivial to show by a simple change of variables in the integral that for Renyi's entropy (as for Shannon's entropy), we have the following identity between the entropies of two scaled random variables.

$$H_\alpha(aY) = H_\alpha(Y) + \log|a| \tag{6.8}$$

where we can replace $\log|a|$ with $(1/2)\log a^2$. We also know that for variance

$$Var(aY) = a^2 Var(Y) \tag{6.9}$$

Combining these two identities, the terms with a cancel out and we get the desired result. In practice, we will use our nonparametric estimator in place of the actual entropy expression in the cost function given in Eq. (6.7). The sample variance estimators already satisfy the scaling property of the variance given in Eq. (6.9), however, we saw that in order for the entropy estimator to satisfy the scaling property of entropy given in Eq. (6.8), we need to scale up the kernel size with the same ratio as the scaling of the norm of the weight vector (assuming WSS observations). Therefore, we may determine a suitable kernel size (performance does not critically depend on this choice as long as the selected kernel size is not very small or very large) corresponding to a unit norm weight vector and initialize the weight vector to be unit norm. Then, during the course of adaptation, we can scale up/down the kernel size according to the new norm of the weight vector.

In Chapter 2, we established that for smooth, symmetric, unimodal kernel functions, the global minima of the nonparametric entropy estimator and the actual entropy coincide and furthermore, this global minimum of the estimator is smooth, i.e., has zero gradient and a positive semi-definite Hessian. This property of the estimator allows gradient and Hessian based optimization procedures to safely converge to the desired optimum point in adaptation.

In addition, Theorem 2.4 allows us to minimize the estimated entropy of the of the data in place of the actual entropy in blind deconvolution, asymptotically (and on the average) the nonparametric estimator results in a larger entropy estimate, we will be minimizing an upper bound for a quantity that we wish to minimize.

Practically, since the deconvolving filter w_n is a causal FIR filter, after the addition of a sufficiently long delay line (length L) due to reasons about the nonminimum phase situation mentioned before, one can express its output as a linear combination of the input samples at consecutive time steps as

$$y_k = w^T X_k \tag{6.10}$$

where the weight vector $w = [w_0 \ldots w_{2L}]^T$ consists of the FIR impulse response coefficients and $X_k = [x_k \ldots x_{k-2L}]^T$ consists of the most recent values of the input signal to the filter.

As for the variance term in Eq. (6.7), under the assumption that the source signal is zero-mean WSS and the unknown channel is linear time-invariant, we can write

$$Var(Y) = Var(X) \cdot \sum_{i=0}^{2L} w_i^2 \tag{6.11}$$

Substituting Eq. (2.5) and Eq. (6.11) in Eq. (6.7), we get the nonparametric estimate of the cost function as

$$\hat{J}(w) = \frac{1}{1-\alpha} \log \frac{1}{N^\alpha} \sum_{j=1}^{N} \left(\sum_{i=1}^{N} \kappa_\sigma (y_j - y_i) \right)^{\alpha-1} - \frac{1}{2} \log \sum_{i=0}^{L} w_i^2 \tag{6.12}$$

where *Var*(*X*) dropped out because it does not depend on the weights of the adaptive filter. Now, using Eq. (6.10), the gradient of the cost function in Eq. (6.12) with respect to the weight vector is obtained as

$$\frac{\partial \hat{J}}{\partial w} = -\frac{\sum_{j=1}^{N} \left(\sum_{i=1}^{N} \kappa_\sigma (y_j - y_i) \right)^{\alpha-2} \left(\sum_{i=1}^{N} \kappa'_\sigma (y_j - y_i)(X_j^T - X_i^T) \right)}{\sum_{j=1}^{N} \left(\sum_{i=1}^{N} \kappa_\sigma (y_j - y_i) \right)^{\alpha-1}} - \frac{w^T}{w^T w} \tag{6.13}$$

where $\kappa'_\sigma(.)$ is the derivative of the kernel function with respect to its argument. Given N samples of X_k, the adaptive filter may be trained to converge to the inverse of the channel. The gradient in Eq. (6.13) may be used in both off-line and on-line training to minimize Eq. (6.12). Choosing a sufficiently small window length of N (depending on the computational requirements), which may be sliding or non-overlapping, it is possible to estimate the source signal on-line.

As for the optimization techniques that can be applied to obtain the optimal solution, simple gradient descent, conjugate-gradient, Levenberg-Marquardt, or other approaches may be taken [Lue73]. If the kernel size is chosen sufficiently large (usually, a kernel width that covers about 10 samples on the average yields good results), then the performance surface is reasonably simple to search and based on numerous simulations, we conjecture that there does not exist local minima. In fact, in all previous problems, it was observed that as long as the kernel size is chosen in a moderate value range (as prescribed above), its precise value is not crucial to the final performance of the adaptive system. An important property of the proposed estimator that deserves mentioning at this point is its close relationship with the global optimization method of convolution smoothing mentioned in Chapter 4.

6.3 Simulations on Minimum Entropy Deconvolution

In order to test the performance of the proposed blind deconvolution algorithm, we performed a series of Monte Carlo runs using different entropy orders and batch-sizes. In the Monte-Carlo runs, a random minimum-phase 15-tap FIR filter is chosen for the unknown channel impulse response, and the length of the deconvolving filter is set to that of the ideal inverse filter. For various values of N and α, 100 random-choice (both for Cauchy distributed data samples and deconvolver initial weights) simulations are run for each combination of (N,α). The results of these Monte Carlo simulations are summarized in Table 6-1 and Table 6-2, where the average and standard deviations of both signal-to-interference-ratio (SIR) and convergence time (T_c) are given.

The SIR of a single run is defined as the average of the SIR values of the last 100 iterations after convergence of that simulation (since due to the constant step size, the performance rattles slightly after convergence). The SIR value at a given iteration is computed as the ratio of the power of the maximum component of the overall filter to the power of the other components, i.e., if we let a_n=h_n*w_n be the overall filter where w_n is the current estimate of the deconvolving filter, we evaluate

$$SIR = 10\log_{10}\frac{[\max(a_i)]^2}{\sum_i a_i^2 - [\max(a_i)]^2} \tag{6.14}$$

Note that under the assumption of WSS source signals, the power of the observed signal is time-invariant; therefore, the overall filter weights can equivalently be used to determine the signal-to-interference ratio. The convergence time is defined as the largest iteration index smaller than the maximum number of iterations minus 100, such that the SIR value is less than or equal to the minimum SIR value attained in the last 100 iterations.

Notice that, regardless of the entropy order and batch size, an average deconvolution performance above 20dB was attained. As expected, with increased batch size, the SIR improved, although slightly. In addition, convergence was mostly achieved within 60 to 70 iterations (note that this number may depend on the step size selected in the gradient descent algorithm). For the specific source distribution, increasing the entropy order seems to decrease the average SIR slightly, but decreases the standard deviation. This conclusion, however, cannot be carried out to general data distributions, yet. In order to determine how the entropy order affects the performance (which we expect to be insignificantly small for most situations) more theoretical and experimental study is required.

Table 6-1. E[SIR] ± std[SIR] in dB over 100 Monte Carlo runs for each combination of batch size and entropy order after convergence

	$\alpha = 1.01$	$\alpha = 2$	$\alpha = 3$
$N = 50$	21.25 ± 1.55	20.49 ± 1.15	20.45 ± 1.09
$N = 75$	21.45 ± 1.45	21.18 ± 1.08	21.09 ± 1.15
$N = 100$	22.14 ± 1.33	21.99 ± 1.04	21.93 ± 1.05
$N = 200$	22.81 ± 1.72	22.49 ± 1.11	22.30 ± 1.17
$N = 300$	22.70 ± 1.81	22.68 ± 1.20	22.65 ± 1.11
$N = 400$	23.04 ± 1.84	22.45 ± 1.48	22.53 ± 1.58
$N = 500$	23.27 ± 2.50	22.71 ± 1.65	22.87 ± 1.75

Table 6-2. E[T_c] ± std[T_c] in iterations over 100 Monte Carlo runs for each combination

	$\alpha = 1.01$	$\alpha = 2$	$\alpha = 3$
$N = 50$	72 ± 22	62 ± 30	62 ± 29
$N = 75$	66 ± 25	64 ± 27	62 ± 29
$N = 100$	65 ± 24	67 ± 28	67 ± 30
$N = 200$	68 ± 25	61 ± 27	62 ± 28
$N = 300$	68 ± 24	64 ± 29	64 ± 30
$N = 400$	70 ± 23	62 ± 29	60 ± 28
$N = 500$	67 ± 25	63 ± 26	64 ± 28

6.4 Constant Modulus Blind Equalization Using Renyi's Entropy

The techniques presented above, namely minimum/maximum entropy deconvolution approaches are useful in cases where the source is a continuous random variable. In digital communications, the transmitted sequence (i.e., the source signal) consists of a known finite set of equiprobable symbols, leading to a discrete probability mass function. In blind equalization of digital communication channels, this additional information may be and is being used. Specifically, in situations where the modulation scheme is based on phase shift keying like BPSK, QPSK, or M-PSK, a family of algorithms known as the constant modulus algorithm (CMA) belonging to the techniques known as Godard-type algorithms is used [God80, Hay94, Hay00b, Tre83]. The main idea behind this technique is to use the knowledge that the source symbols, which are complex-valued, have a constant magnitude and, therefore one can adapt the equalizer weights to minimize the error between the output symbol moduli and the known constant modulus of the source symbols. When this error criterion is taken to be MSE, one obtains the commonly used forms of the CMA algorithm [Hay00b]. Motivated by the CM-principle and inspired by our minimum error entropy algorithm, Santamaria *et al.* has recently proposed an extension to the family of CMA [San02b]. In their paper, basically they replace the MSE criterion in the CMA approach with the MEE criterion applied to the square of the modulus of the equalizer output symbols. They normalize the weight vector after each update by setting the center-tap to one and compare the performance of the entropy-based CMA with its MSE-based counterpart to arrive at conclusions in favor of the entropy-based algorithm. In this section, we will present the formulation for this blind equalization algorithm. In order to avoid the requirement of weight scaling at every iteration, we will make use of the scale invariant form given in Eq. (6.7). Finally, we will present simulation results from the blind equalization of BPSK and QPSK symbol sequences using this algorithm.

Suppose the channel under consideration is an LTI, FIR filter with complex tap-weights in general, since we use the baseband representation for the transmitted signal. Let s_k be the (complex-valued) symbol transmitted at time k and let n_k be the zero-mean additive white Gaussian noise (AWGN) acting on the received signal. Then the received signal at time k is given by the convolution of the channel impulse response and the input symbol sequence plus the additive noise.

$$x_k = \sum_{l=0}^{L_h-1} h_l s_{k-l} + n_k \tag{6.15}$$

The output of our FIR equalizer (assuming minimum-phase channel and causal equalizer without loss of generality) is similarly given by

$$y_k = \sum_{l=0}^{L_w-1} w_l x_{k-l} = W^T X_k \tag{6.16}$$

where we defined the vectors $w \overset{\Delta}{=} [w_0 \quad \cdots \quad w_{L_w-1}]^T$ and $X_k \overset{\Delta}{=} [x_k \quad \cdots \quad x_{k-L_w+1}]^T$ for notational convenience. Notice that s_k, x_k, and y_k are complex-valued in general.

In the Godard-type CM algorithms, the equalizer weight vector is optimized to minimize the error between $|\, y_k \,|^p$ and $R_p \overset{\Delta}{=} E[|\, s_k \,|^{2p}]/E[|\, s_k \,|^p]$. In this context, one could also minimize the entropy of this error, leading to the following family of cost functions.

$$J_\alpha^p(w) = H_\alpha(|\, y \,|^p - R_p) \tag{6.17}$$

Since entropy is invariant to the mean of its argument pdf, the computation of R_p is not necessary in Eq. (6.17) and $J_\alpha^p(w) = H_\alpha(|y|^p)$ could alternatively be minimized. Finally, in order to make this a scale invariant cost function, we introduce the logarithm of the variance of the random variable of interest to obtain finally

$$J_\alpha^p(w) = H_\alpha(|y|^p) - \frac{1}{2}\log Var[|y|^p] \tag{6.18}$$

Once again, we will substitute our entropy estimator in Eq. (6.18) together with the sample variance estimate to obtain the nonparametric estimator for this cost function, assuming that a sample batch size of *B* is used. For further consideration, we will assume that the family parameter $p = 2$.

$$\begin{aligned} \hat{J}_\alpha^2(w) &= \frac{1}{1-\alpha}\log\frac{1}{B^\alpha}\sum_{j=1}^{B}\left(\sum_{i=1}^{B}\kappa_\sigma(|y_j|^2 - |y_i|^2)\right)^{\alpha-1} \\ &\quad -\frac{1}{2}\log\left(\frac{1}{B}\sum_{j=1}^{B}|y_j|^2 - \left(\frac{1}{B}\sum_{j=1}^{B}|y_j|\right)^2\right) \end{aligned} \tag{6.19}$$

Noting that the gradient of the modulus of the equalizer output with respect to the weight vector can be expressed simply as (notice that the actual gradient is the transpose of what is presented below)

$$\begin{aligned} \frac{\partial |y_k|}{\partial w} &= \frac{1}{2|y_k|}\left[y_k X_k^* + y_k^* X_k\right] \\ \frac{\partial |y_k|^2}{\partial w} &= y_k X_k^* + y_k^* X_k \end{aligned} \tag{6.20}$$

we can easily determine the gradient of Eq. (6.19) with respect to the weight vector to be as in Eq. (6.21). Notice that everything necessary to evaluate the second term of the gradient in Eq. (6.21) already needs to be computed to evaluate the first term. Therefore, the introduction of this term contributes *O*(*B*) number of additions and a few additional multiplication and divisions to the computational complexity of the gradient, which has already a complexity of $O(B^2)$ due to the double summations of the first term.

$$\frac{\partial \hat{J}_\alpha^2(w)}{\partial w} = -\frac{\sum_{j=1}^{B}\left(\sum_{i=1}^{B}\kappa_\sigma(|y_j|^2-|y_i|^2)\right)^{\alpha-2}\left(\sum_{i=1}^{B}\begin{array}{c}\kappa'_\sigma(|y_j|^2-|y_i|^2)\cdot\\ \left(\frac{\partial|y_j|^2}{\partial w}-\frac{\partial|y_i|^2}{\partial w}\right)\end{array}\right)}{\sum_{j=1}^{B}\left(\sum_{i=1}^{B}\kappa_\sigma(|y_j|^2-|y_i|^2)\right)^{\alpha-1}} - \frac{1}{2}\frac{\sum_{j=1}^{B}\frac{\partial|y_j|^2}{\partial w}-\frac{2}{B}\left(\sum_{j=1}^{B}|y_j|\right)\cdot\left(\sum_{j=1}^{B}\frac{\partial|y_j|}{\partial w}\right)}{\sum_{j=1}^{B}|y_j|^2-\frac{1}{B}\left(\sum_{j=1}^{B}|y_j|\right)^2} \tag{6.21}$$

The gradient expression in Eq. (6.21) can be used for both off-line and on-line blind equalization. In the on-line case, the batch size B can be chosen to be a suitably large number that does not exceed the allocated computational resources for this task. Two approaches may be followed: In the sliding window approach, the data batch of size B can be updated with every new incoming sample by discarding the oldest data point and in the non-overlapping windows approach for every batch a new set of B samples can be taken. In any case, the algorithm that trains the weights using standard gradient descent will converge to the optimal solution on the average.

We investigated the performance of this algorithm on the blind deconvolution of BPSK and QPSK modulation schemes. For practicality, the batch size must be chosen to be small, however, this caused the sample estimate of the variance term in the cost function to be inadequate for a robust adaptation process. Therefore, we performed training without this additional term. As an illustration of the results we obtained, we present the convergence plots of a length five equalizer, where the *unknown* channel is an all-pole system with four real poles at {0.1,0.3,0.5,0.7} and the modulation scheme is QPSK. In the implementation of this IIR channel, we truncated the impulse response after index 27 because the energy concentrated in the remaining terms was insignificant. The following figure summarizes these results.

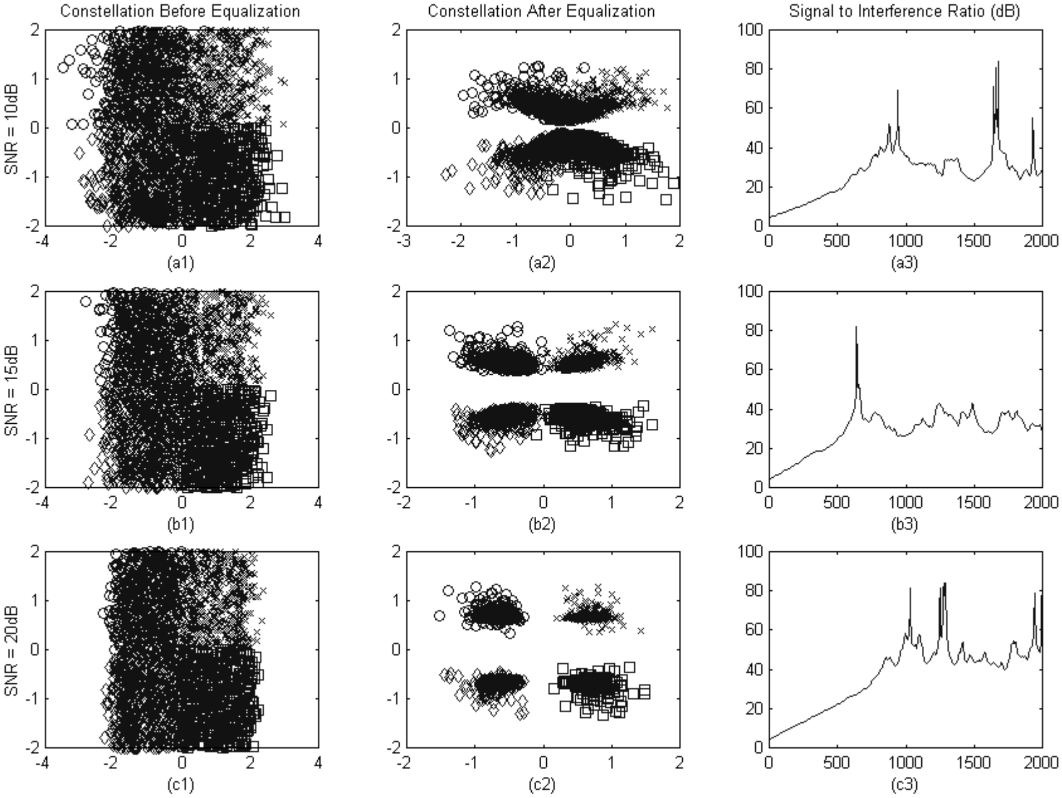


Figure 6-2. A sample from the simulations with entropy-based constant modulus blind equalization algorithm for QPSK modulation. Each row of subfigures corresponds to SNR levels of 10dB, 15dB, and 20dB, respectively. Each column of subfigures shows signal constellations before equalization, after equalization and SIR versus iterations

Notice that as the signal-to-noise ratio (SNR) measured at the receiver input increases the accuracy of the blind equalization algorithm also improves, as expected. The SIR measure used in the third column of subfigures is the negative of the commonly used inter-symbol interference (ISI) performance measure for equalizers. Although, in all the tried noise levels, the equalizer learned the inverse of the channel in the range of 20dB to 40dB on the average, the additive noise is not targeted intentionally. However, constant modulus algorithms, in general, are experimentally shown to converge to the vicinity of the minimum MSE solution [Hay00b]. Therefore, we can expect that the entropy-based CMA also achieves some level of noise reduction. This expectation is also supported by the asymptotic noise rejection capability of the MEE criterion.

Santamaria *et al.* also investigated, although not thoroughly, the effect of entropy order and batch size on the performance of this algorithm. Their results indicated that these considerations were insignificant and little performance change has been observed [San02b].

In this chapter, we described briefly the blind deconvolution problem and proposed the use of Renyi's entropy in solving this problem using two approaches, namely minimum and maximum entropy deconvolution. We outlined the principles for the maximum entropy deconvolution algorithm and presented a general-purpose minimum entropy deconvolution algorithm. Investigations of this algorithm in solving blind deconvolution problems with continuous- and discrete-valued source signals showed its effectiveness.

CHAPTER 7
STOCHASTIC INFORMATION GRADIENT

7.1 Stochastic Approximation to the Gradient of Entropy

In the preceding chapters, we presented a nonparametric estimator for Renyi's entropy, based on Parzen windowing and the resubstitution technique. Successful applications of this estimator to numerous adaptation problems were showed. In all these applications, we used a batch of training data points when evaluating the weight updates at each iteration, regardless of whether the training progressed off-line or on-line. In real-time on-line adaptation processes, the computational complexity of the learning algorithm must be reasonably feasible for practical implementation. In the first years of adaptive filtering with digital computers the same problem was encountered and addressed by researchers. Widrow's stochastic gradient for the MSE criterion for the training of FIR filters, which led to the celebrated LMS algorithm, has proven extremely effective and efficient; subsequently this algorithm not only survived the decades, but it has also become possibly the most popular adaptation algorithm in the literature and in practice [Far98, Hay84, Hay96, Wid85]. In order for practical applicability of the proposed algorithm, which is $O(N^2)$ complexity for N samples, in real-time adaptation scenarios, we are compelled to determine a simplification to the weight updates when training the adaptive system under a performance measure based on entropy, specifically our estimator. To this end, we will assume the strategy used by Widrow when he was deriving the stochastic gradient for the MSE criterion. The details of our derivation for the stochastic gradient for entropy, which we will name as the *stochastic information gradient* (SIG) are presented in this chapter, along with successful applications in various learning scenarios.

7.1.1 Stochastic Gradient for Shannon's Entropy

Recall Shannon's entropy for a random variable Y with pdf $f_Y(y)$ [Cov91, Sha64]

$$H_S(Y) = -\int_{-\infty}^{\infty} f_Y(y) \log f_Y(y) dy = E_Y\left[-\log f_Y(Y)\right] \tag{7.1}$$

As in the nonparametric estimator for Renyi's entropy, we can estimate the *unknown* pdf of the variable under consideration from its samples, $\{y_1, \ldots, y_N\}$ using Parzen windowing.

$$\hat{f}_Y(y) = \frac{1}{N}\sum_{j=1}^{N} \kappa_\sigma(y - y_j) \tag{7.2}$$

Similar to Eq. (2.5), replacing the expectation in Eq. (7.1) with the sample mean and substituting the Parzen pdf estimate, one could obtain a nonparametric estimate of Shannon's entropy.

As in Widrow's approach in deriving LMS, where he dropped the expectation operator in the theoretical MSE definition and approximated this quantity with the instantaneous value of the error-square, we will drop the expectation from the definition of Shannon's entropy and use the most current sample of Y in the pdf to obtain the following stochastic estimate for entropy: $H_S(Y) = E_Y[-\log f_Y(Y)] \approx -\log f_Y(y_k)$. In this, y_k denotes the most recent sample of Y at time step k. Since, in practice, the pdf of Y is unknown, we will use the estimate in Eq. (7.2) evaluated over the most recent L samples of Y, resulting in the following instantaneous estimate of the pdf evaluated at y_k.

$$\hat{f}_Y(y_k) = \frac{1}{L}\sum_{i=k-L}^{k-1} \kappa_\sigma(y_k - y_i) \tag{7.3}$$

Thus the stochastic estimate of Shannon's entropy at time *k* becomes

$$\hat{H}_{S,k}(Y) = -\log\left(\frac{1}{L}\sum_{i=k-L}^{k-1}\kappa_\sigma(y_k - y_i)\right) \tag{7.4}$$

It can easily and clearly be seen that the expected value of Eq. (7.4) satisfies

$$E[\hat{H}_{S,k}(Y)] = E[-\log \hat{f}_Y(y_k)] = \hat{H}_S(Y) \tag{7.5}$$

where $\hat{H}_S(Y)$ is Shannon's entropy estimated using Parzen windows. Now that we have a stochastic estimate of entropy at every time instant, we can easily determine its gradient. Assuming that the samples are generated by an adaptive system with weight vector *w*, this stochastic entropy gradient is found to be

$$\frac{\partial \hat{H}_{S,k}}{\partial w} = -\frac{\sum_{i=k-L}^{k-1}\kappa'_\sigma(y_k - y_i)\left(\frac{\partial y_k}{\partial w} - \frac{\partial y_i}{\partial w}\right)}{\sum_{i=k-L}^{k-1}\kappa_\sigma(y_k - y_i)} \tag{7.6}$$

where the length *L* of the sliding window could be selected in consideration with the length of the duration where the samples can be assumed iid (notice that in practice the whole estimation process depends on the samples being iid, which is most likely not the case in adaptation; specifically either the independence or the identicalness portion of the assumption may become invalid; however all simulations proved that this violation of the assumptions does not cause problems in practice as the nonparametric estimator itself starts behaving as a suitable finite-sample case cost function in all applications). The expression in Eq. (7.6) is called the *stochastic information gradient*. The trade-off here in selecting *L* is between tracking capability and misadjustment. Clearly, the expected value of this gradient is equal to the actual gradient of Shannon's entropy estimated by Parzen windowing, as a direct consequence Eq. (7.5), assuming that the derivation can be interchanged with the expectation operator. This SIG was successfully used in solving BSS and BD problems; these applications will be discussed in the following sections.

7.1.2 Stochastic Gradient for Order-α Information Potential

A similar approach could be taken in order to determine a stochastic gradient for the information potential, thus Renyi's entropy. Recall that, we defined the information potential of a random variable Y as

$$V_\alpha(Y) = \int_{-\infty}^{\infty} f_Y^\alpha(y)dy = E_Y[f_Y^{\alpha-1}(Y)] \tag{7.7}$$

Once again, dropping the expectation and stochastically approximating the value of this operation with the instantaneous value of its argument, we obtain $V_\alpha(Y) \approx f_Y^{\alpha-1}(y_k)$. Now, substituting the Parzen window estimate for the pdf, we obtain the stochastic estimate for order-α information potential as

$$\hat{V}_{\alpha,k}(Y) = \left(\frac{1}{L} \sum_{j=k-L}^{k-1} \kappa_\sigma(y_k - y_j) \right)^{\alpha-1} \tag{7.8}$$

Consequently, the stochastic gradient is easily determined to be

$$\begin{aligned} \frac{\partial \hat{V}_{\alpha,k}(Y)}{\partial w} = (\alpha-1) \cdot & \left(\frac{1}{L} \sum_{j=k-L}^{k-1} \kappa_\sigma(y_k - y_j) \right)^{\alpha-2} \\ & \cdot \left(\frac{1}{L} \sum_{j=k-L}^{k-1} \kappa'_\sigma(y_k - y_j) \cdot \left(\frac{\partial y_k}{\partial w} - \frac{\partial y_j}{\partial w}\right) \right) \end{aligned} \tag{7.9}$$

Clearly, the expected value of both Eq. (7.8) and Eq. (7.9), due to the same reasoning as in the previous section, equal their corresponding nonparametric estimates using the whole data set. Recalling that Renyi's entropy is defined as $H_\alpha(Y) = [\log V_\alpha(Y)]/(1-\alpha)$ and that its gradient with respect to the weights is given in terms of the information potential and its gradient by $\partial H_\alpha(Y)/\partial w = [\partial V_\alpha(Y)/\partial w]/[(1-\alpha)\cdot V_\alpha(Y)]$, we substitute Eq. (7.8) and Eq. (7.9) in this to obtain the stochastic gradient for Renyi's entropy.

$$\frac{\partial \hat{H}_{\alpha,k}(Y)}{\partial w} = \frac{1}{1-\alpha} \frac{\left[\begin{array}{r} (\alpha-1)\cdot\left(\frac{1}{L}\sum\limits_{j=k-L}^{k-1}\kappa_\sigma(y_k-y_j)\right)^{\alpha-2} \\ \cdot\left(\frac{1}{L}\sum\limits_{j=k-L}^{k-1}\kappa'_\sigma(y_k-y_j)\cdot(\frac{\partial y_k}{\partial w}-\frac{\partial y_j}{\partial w})\right) \end{array} \right]}{\left(\frac{1}{L}\sum\limits_{j=k-L}^{k-1}\kappa_\sigma(y_k-y_j)\right)^{\alpha-1}} \tag{7.10}$$

which, after some cancellations, simplifies down to

$$\frac{\partial \hat{H}_{\alpha,k}}{\partial w} = -\frac{\sum\limits_{i=k-L}^{k-1}\kappa'_\sigma(y_k-y_i)\left(\frac{\partial y_k}{\partial w}-\frac{\partial y_i}{\partial w}\right)}{\sum\limits_{i=k-L}^{k-1}\kappa_\sigma(y_k-y_i)} = \frac{\partial \hat{H}_{S,k}}{\partial w} \tag{7.11}$$

Interestingly, in the stochastic gradient for Renyi's entropy, the entropy order disappears and the resulting gradient expressions turn out to be identical to that of Shannon's entropy. This should not confuse us; in fact, we know that ideally when compared, the entropies of two arbitrary pdfs are ordered in the same way whether Shannon's or Renyi's definition is used. Therefore, (asymptotically speaking) it is only natural for the gradients to point towards the same direction, converging to the same optimal solution. This SIG is successfully used by Erdogmus and Principe [Erd01b, Erd01c] to solve supervised adaptation problems using the MEE criterion.

7.1.3 Alternative Stochastic Gradient Expressions

Although we considered approximating the pdf using a window of L samples and reducing the expectation to a single value evaluation of its argument, we could have taken other approaches. For example, we could approximate the expectation with a sample mean over the most recent L samples, while estimating the pdf using only one sample from the past. In that case, the stochastic Shannon's entropy estimate would be

$$\hat{H}_{S,k}(Y) = -\frac{1}{L}\sum_{j=k-L+1}^{k}\log\kappa_{\sigma}(y_j - y_{j-1}) \tag{7.12}$$

In this case, the stochastic gradient would become

$$\frac{\partial \hat{H}_{S,k}(Y)}{\partial w} = -\frac{1}{L}\sum_{j=k-L+1}^{k}\frac{\kappa'_{\sigma}(y_j - y_{j-1})\cdot(\frac{\partial y_j}{\partial w} - \frac{\partial y_{j-1}}{\partial w})}{\kappa_{\sigma}(y_j - y_{j-1})} \tag{7.13}$$

For this approach, we would determine the stochastic information potential estimate and its gradient to be as follows.

$$\hat{V}_{\alpha,k}(Y) = \frac{1}{L}\sum_{j=k-L+1}^{k}\kappa_{\sigma}^{\alpha-1}(y_j - y_{j-1}) \tag{7.14}$$

$$\frac{\partial \hat{V}_{\alpha,k}(Y)}{\partial w} = \frac{\alpha-1}{L}\sum_{j=k-L+1}^{k}\kappa_{\sigma}^{\alpha-2}(y_j - y_{j-1})\cdot\kappa'_{\sigma}(y_j - y_{j-1})\cdot(\frac{\partial y_j}{\partial w} - \frac{\partial y_{j-1}}{\partial w}) \tag{7.15}$$

Substituting these to determine the stochastic gradient for Renyi's entropy, we obtain

$$\frac{\partial \hat{H}_{\alpha,k}(Y)}{\partial w} = -\frac{\sum_{j=k-L+1}^{k}\kappa_{\sigma}^{\alpha-2}(y_j - y_{j-1})\cdot\kappa'_{\sigma}(y_j - y_{j-1})\cdot(\frac{\partial y_j}{\partial w} - \frac{\partial y_{j-1}}{\partial w})}{\sum_{j=k-L+1}^{k}\kappa_{\sigma}^{\alpha-1}(y_j - y_{j-1})} \tag{7.16}$$

This SIG has been used successfully by Hild *et al.* [Hil01b, Hil02] to solve the BSS problem on-line. This application will be discussed in the following sections in more detail with simulation results.

Obviously, the main advantage of all the SIG expressions is to decrease the complexity of the summations from $O(N^2)$ to $O(L)$, where N is the total number of samples in the training data. Despite this tremendous reduction of complexity, the performance of SIG-based learning algorithms are virtually comparable with those of the batch-mode algorithms that use the whole training set in every update computation. However, convergence to a unique solution without any fluctuations should not be expected; as in any stochastic gradient algorithm, SIG results in some misadjustment and variation about the optimal solution. Conversely, this rattling effect of SIG could be advantageous in some situations; it could help the algorithm to escape from the domain of attraction of any *small* local optima to result in global optimization, in surfaces where any such sub-optimal solutions might exist.

7.2 Relationship Between SIG and Hebbian Learning

Consider an extreme special case of the stochastic gradient given in Eq. (7.6), where the window length L is taken as one. Denoting the gradient of an output sample y_k with respect to the weight vector (i.e., the sensitivity) by $S_w(x_k) = \partial y_k / \partial w$, which is a function of the corresponding input x_k, we obtain this special-case SIG as

$$\frac{\partial \hat{H}_{\alpha,k}}{\partial w} = g(y_k - y_{k-1}) \cdot (S_w(x_k) - S_w(x_{k-1})) \tag{7.17}$$

where $g(x) \overset{\Delta}{=} -\kappa'_\sigma(x) / \kappa_\sigma(x)$ is, in general, a nonlinear function defined by the selected kernel function. For the specific choices of Gaussian kernels and an ADALINE structure ($y_k = w^T x_k$) for which the output is a linear combination of the input components, Eq. (7.17) further reduces to the simple form given in Eq. (7.18) (writing the gradient as a column vector).

$$\frac{\partial \hat{H}_{\alpha,k}(Y)}{\partial w} = \frac{1}{\sigma^2} (y_k - y_{k-1}) \cdot (x_k - x_{k-1}) \tag{7.18}$$

We notice that this update rule (the gradient multiplied by a learning rate) resembles the classical Hebbian updates (given by y_kx_k in this context) in the neural networks literature [Hay99], where the Hebbian rule stated 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”. It is well known that Hebbian updates manipulate the correlation (maximize or minimize depending on the sign of the updates) between the output signal and the inputs and maximize (or minimize) the output variance. We derived Eq. (7.18), however, starting from the entropy of the output, and showed that on the average these updates maximize (or minimize) the entropy. It is remarkable that switching from Hebbian updates applied to instantaneous values of the input and output samples to Hebbian rule applied to the instantaneous increments of the signals, it is possible to switch between manipulation of variance and entropy.

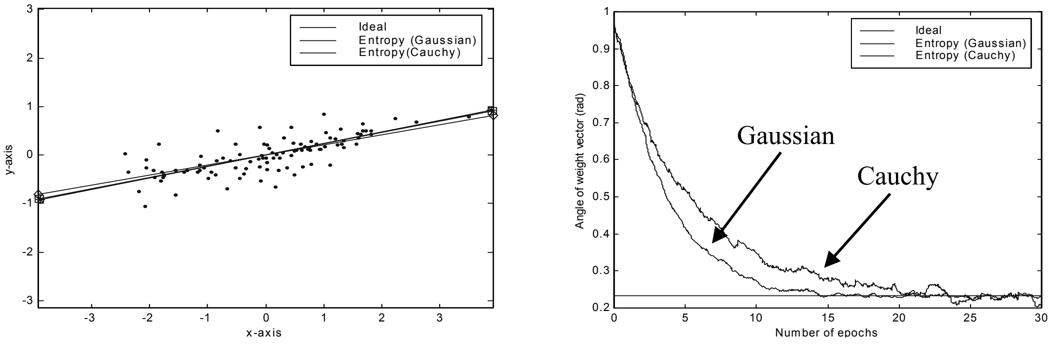
In fact, the special case of Gaussian kernels is not the only one to obey Hebb’s original description of the process. Hebb’s rule states: “When an axon of cell A is near enough to excite cell B or repeatedly or consistently takes part in firing it, some growth or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased” [Heb49]. According to this, the product of the input and the output is not the only configuration that is possible. Notice that the function $g(.)$ satisfies $sign(g(x)) = sign(x)$ when unimodal, symmetric and differentiable kernels are used. Thus, all updates of the form $g(y_k)x_k$ would be feasible Hebbian updates as well. In the context of instantaneous increments, this would become $g(y_k - y_{k-1}) \cdot (x_k - x_{k-1})$, which is the update rule we would obtain for the ADALINE structure for an *arbitrary* choice of the kernel function (the regular limitations mentioned above apply).

Thus, we conclude that when Hebbian learning is applied to the instantaneous differential increments of the output and the input of the ADALINE instead of the instantaneous values of these quantities, the neuron implements information learning rather than merely correlation learning.

In order to show the performance of Eq. (7.17) in determining the maximum entropy direction of a given data set, we present here two simulations. For this purpose, we use Gaussian and Cauchy kernels given by

$$\begin{aligned} G_\sigma(x) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \\ C_\sigma(x) &= \frac{1}{\pi \cdot \sigma} \cdot \frac{1}{1+(x/\sigma)^2} \end{aligned} \tag{7.19}$$

After each update, the weight vector of the ADALINE is normalized to prevent from diverging; when the weight is normalized, the *previous output* is also modified accordingly for consistency. In the first example, 100 samples from a 2-dimensional joint Gaussian distribution are generated. The learning rates for Gaussian and Cauchy kernels are selected to be 10^{-5} and 10^{-3}, respectively. Both kernel sizes are chosen as σ= 0.1.

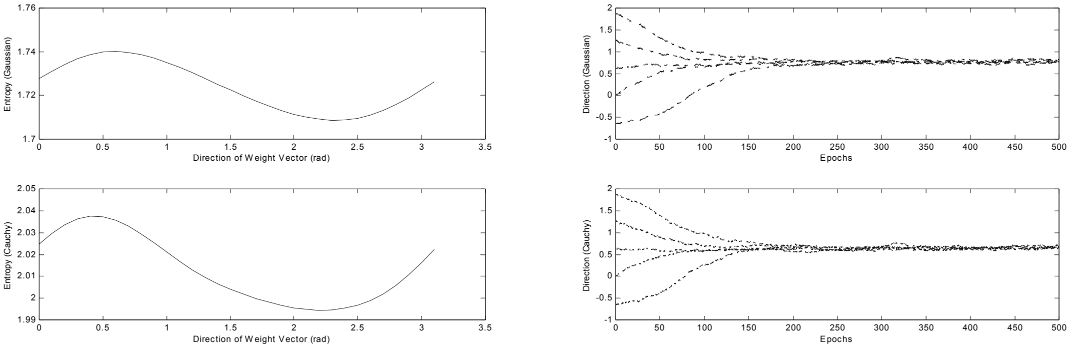


(a) (b)

Figure 7-1. Finding the maximum entropy direction with SIG a) Data samples and the derived directions using both kernels b) Convergence of the angles to the true value determined from the theoretical covariance matrix

Figure 7-1a shows the samples generated along with the final estimated directions using the two different kernel functions. Figure 7-1b shows the angle of the weight vector converging to the ideal solution, which corresponds to the 1st principal component in this case (since the data is Gaussian, maximum entropy and maximum variance directions coincide). The choice of kernel size and learning rate are important in that they affect the convergence time and the magnitude of the fluctuations around the solution.

Our second example we use 50 samples generated from a random vector with independent components, where the x-component is uniformly distributed and the y-component is Gaussian. The covariance matrix of the data is normalized to identity, thus PCA algorithms are unable to deduce any maximal variance direction. On the other hand, using the maximum entropy approach, we can determine the line that maximizes the entropy of the projections (i.e., the direction of maximum uncertainty).



(a) (b)

Figure 7-2. Convergence to maximum entropy direction from different initial conditions a) Estimated entropy vs. direction b) Convergence to the maximum-entropy direction from different initial conditions for the weight vector

Figure 7-2a shows the estimated entropy of the projections versus the direction of the projected line and Figure 7-2b shows the convergence of the weights from different initial conditions. We remark that, in this example, as the number of samples increase, the estimated distributions will converge to the actual uniform-Gaussian distributions. Hence, asymptotically, the estimated direction will converge to $\pi/2$, i.e., the Gaussian direction, since Gaussian has the largest entropy among fixed variance distributions [Cov91].

There are two main conclusions of this section. The first is the possibility that biological neuron assemblies may as well be utilizing entropy rather than mere correlation as the common interpretation of Hebb's Law states; under this hypothesis, Hebbian learning is no longer synonymous with correlation/variance-based learning. The second is that, with a simple modification of the classical Hebbian update, it becomes possible to train artificial neural networks for the optimization of entropy, and perhaps other information theoretic quantities. The proposed formulation has two advantages: It is computationally very simple, specifically on the same order of complexity with the traditional Hebbian learning rule and secondly it extracts more than only the second-order statistical information from the samples.

7.3 Applications of SIG to Supervised and Unsupervised Learning Schemes

The SIG expressions we presented in the preceding sections can be used in many contexts of adaptation, where entropy is an integral part of the cost function. In this section, we will show the performance of these stochastic gradients in determining the solutions of blind deconvolution and blind source separation problems on-line.

In our first case study, we will consider the maximum entropy blind deconvolution scheme that has been described in Chapter 6. In this scheme, recall that we assume we know the pdf of an iid sequence of input samples to an unknown channel from which we get our observations. The deconvolver consists of an FIR filter followed by a nonlinearity that is matched to the cdf of the source signal. In our example, for convenience we use a minimum-phase FIR channel (50taps), whose ideal inverse is a two-tap causal FIR. The distribution of the input signal is Cauchy. We use the SIG in Eq. (7.6) with $L=10$ and Gaussian kernels and a learning rate of $\eta=10^{-3}$. Defining the SIR as in Chapter 6 (i.e., the power of the maximum component of the overall filter impulse response divided by the total power of all the other residual components), we run a simulation whose results are presented in Figure 7-3. Notice that after 10^4 samples, the algorithm achieves an average SIR of 25-30dB. Fine-tuning the window length and the learning rate, the solution may be improved.

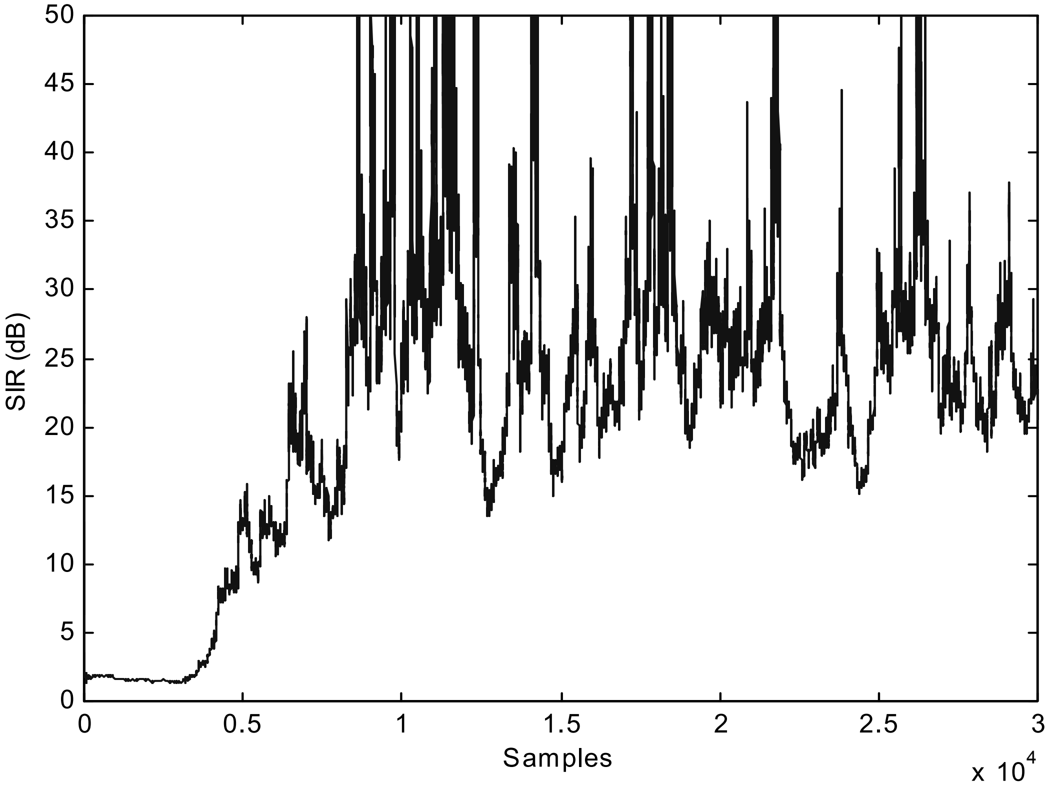


Figure 7-3. SIR versus samples for the on-line maximum entropy blind deconvolution scheme using SIG

Our second case study is on MEE learning in linear adaptive systems (ADALINE) using SIG. For this purpose, we performed two simulations using different input-output data sets. In the first training set, the purpose is to learn the weight vector of a 2-tap FIR filter which will be used to predict the next sample in a sequence generated by sampling the signal $x(t) = \sin 20t + 2\sin 40t + 3\sin 60t$ at 100Hz. The training data consists of 32 input-output pairs, which corresponds to one period of the signal, that is shown to the adaptive FIR filter repeatedly for 150 epochs.

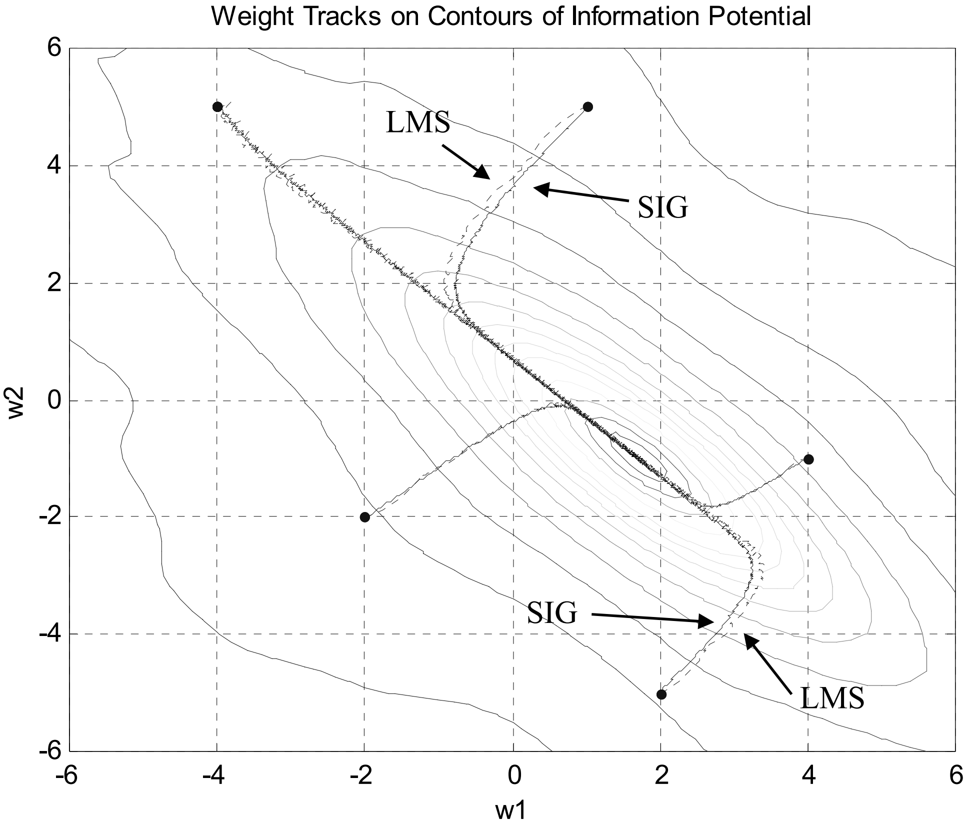


Figure 7-4. Weight tracks for SIG (solid) and LMS (dotted) on the contour plot of the error entropy cost function for the sinusoid-prediction example

For comparison, the FIR filter is trained both using SIG (step size 0.1) and LMS (step size (0.001). With these choices, the convergence of both algorithms is achieved within the given number of iterations. Figure 7-4 depicts the convergence of both algorithms to the optimal solution starting from five different initial conditions of the weight vector. For fixed convergence time, SIG is observed to converge smoother than LMS, however, such a strong conclusion about the misadjustment of the final solution cannot be inferred just by investigating a few simulations; a deeper theoretical investigation is necessary [Erd01c].

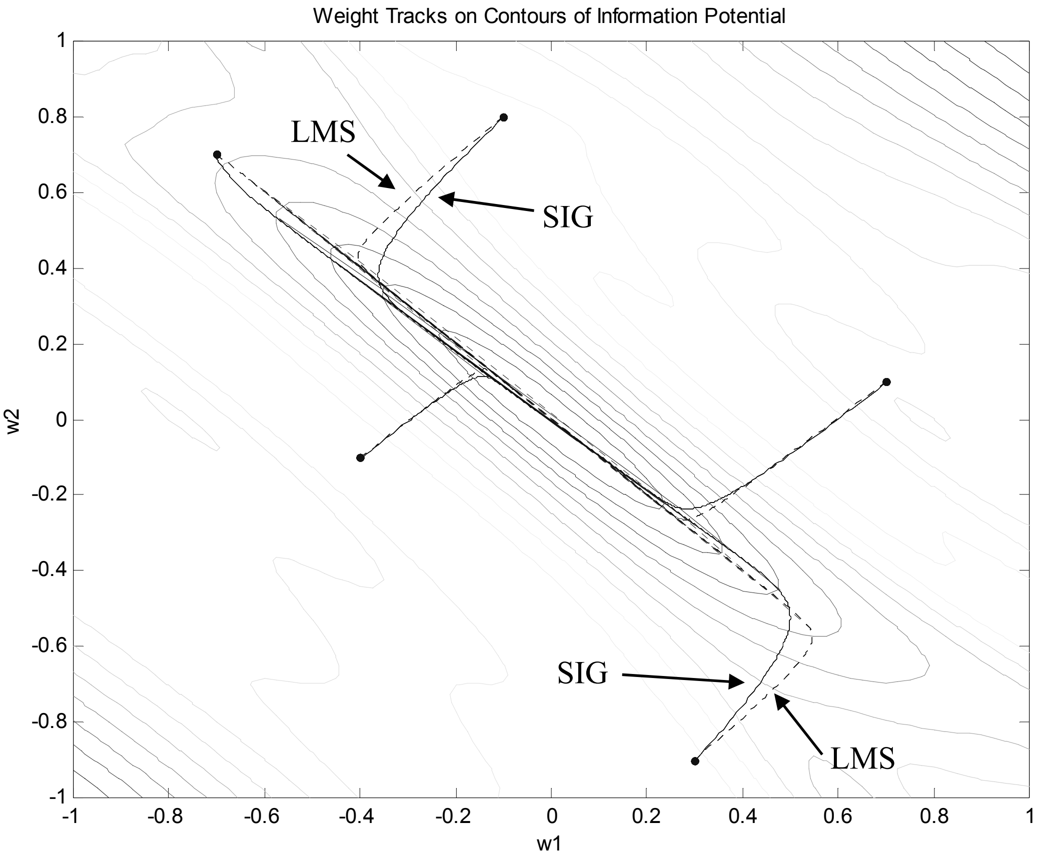


Figure 7-5. Weight tracks for SIG (solid) and LMS (dotted) on the contour plot of the error entropy cost function for the frequency-doubling example

The second example is the frequency-doubling problem, where a 2-tap FIR filter is trained to output a sinusoid with double the frequency of the one at its input. The motivation for this data set is to investigate the behavior of SIG and the structure of the MEE surface in a situation where we know the optimal solution has large approximation errors. Once again, the FIR filter is trained using both SIG and LMS algorithms starting from five initial conditions. The convergence of the weight tracks to the optimal solution is shown in Figure 7-5. The two small equilevel loops on either edge of the performance surface are local maxima, and there is only one local minimum of the entropy, which corresponds to the optimal solution. In this example, the training set size was 20 and the data was presented to the algorithm for 1000 epochs repeatedly [Erd01c].

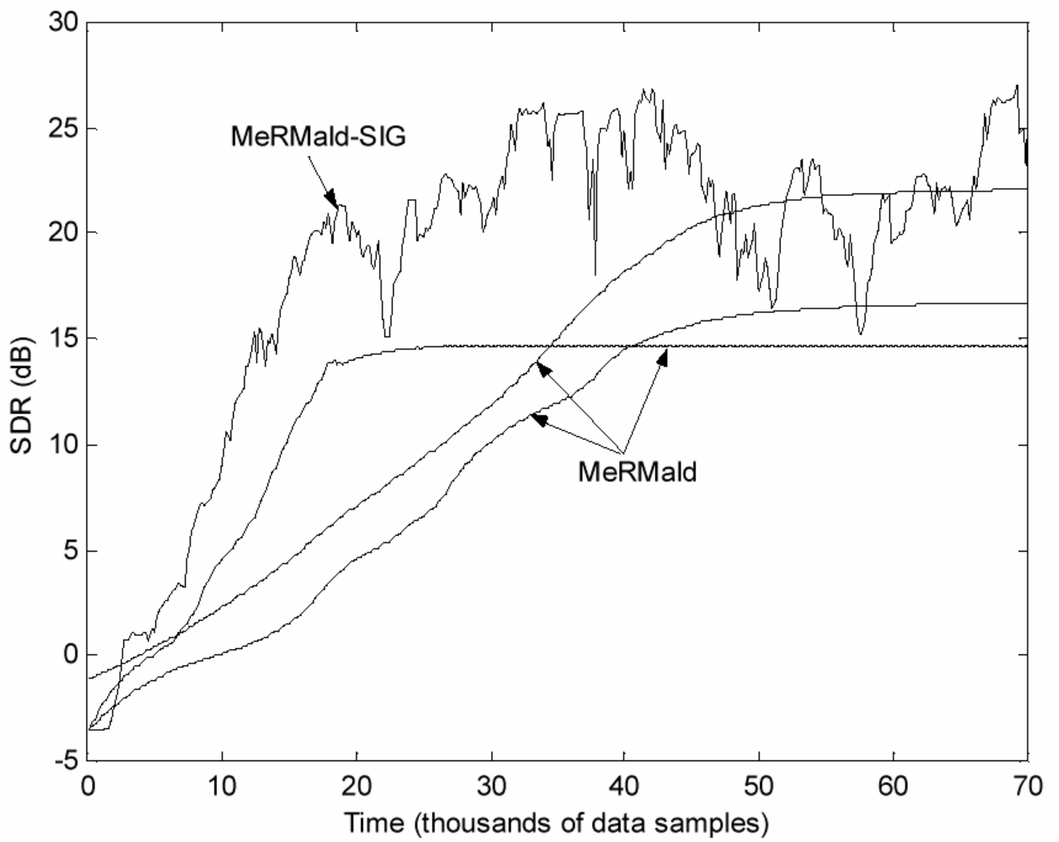


Figure 7-6. SDR versus number of samples seen by the MeRMaId and MeRMaId –SIG algorithms. Each iteration of MeRMaId is considered as 200 samples seen

Our third case study is on blind source separation and it aims to show the effectiveness of SIG in solving this problem. Consider the BSS topology and criterion we suggested in Chapter 5. The topology consisted of a whitening stage followed by Givens rotations, whose angles were optimized according to the *minimize sum of output marginal entropies* principle. In this example, we will assume that the ideal spatial whitening matrix is known *a priori* (that spheres the whole training data set). This way, we will have the opportunity to study the performance of *only* SIG, not influenced by the performance of the specific on-line whitening scheme that would be used otherwise. In addition, we will use the SIG expression given in Eq. (7.16) with entropy order equal to two, for a change. As the performance measure, we will use the SDR measure defined in Chapter 5. In this set of simulations, there are 10 mixtures of 10 audio sources, which comprise of five male speakers, four female speakers, and a piece from a symphony. The mixing coefficients were chosen uniformly in the interval [-1,1]. In the first set of results, we compare the performance of MeRMaId-SIG with the batch-mode MeRMaId (notice that by this time, we changed the name of our BSS algorithm from MRMI to MeRMaId). In the SIG approach, the window length was set to L=200, and the batch mode training, 3 different randomly selected sets of 200 samples were used to train the Givens angles. The results are shown in Figure 7-6 [Hil01b].

Notice that the performance of the batch approach may be susceptible to the particular set of selected training samples. On the other hand, SIG, with its computational simplicity, allows us to train a network in a reasonable amount of time using larger data sets. The trade-off here is between final average performance and the magnitude of fluctuations around this solution, as clearly seen from Figure 7-6. Another set of results, aimed to compare the performance of our MeRMaId-SIG algorithm with other on-line BSS algorithms that are considered benchmarks in the field. The experimental set-up is designed to be consistent with a real-time operation, where each algorithm will see each data sample once, use it to compute weight updates, and discard it. In this comparison, we do not include Comon’s MMI and Hyvarinen’s FastICA, because they are essentially batch algorithms. Although some modifications could be made towards using them in on-line schemes, their performances would not be comparable. We, therefore, limit the comparison to Bell-Sejnowski’s InfoMax and Yang’s MMI algorithms (both use Amari’s natural gradient [Ama96, Ama98]). The signals to be separated are two audio sources, selected from the group of 10 mentioned above. At 16.384 KHz, the length of the data shown below is about 8.5 seconds. The first 0.5 seconds of both recordings is silence (some noise), therefore adaptation does not immediately occur.

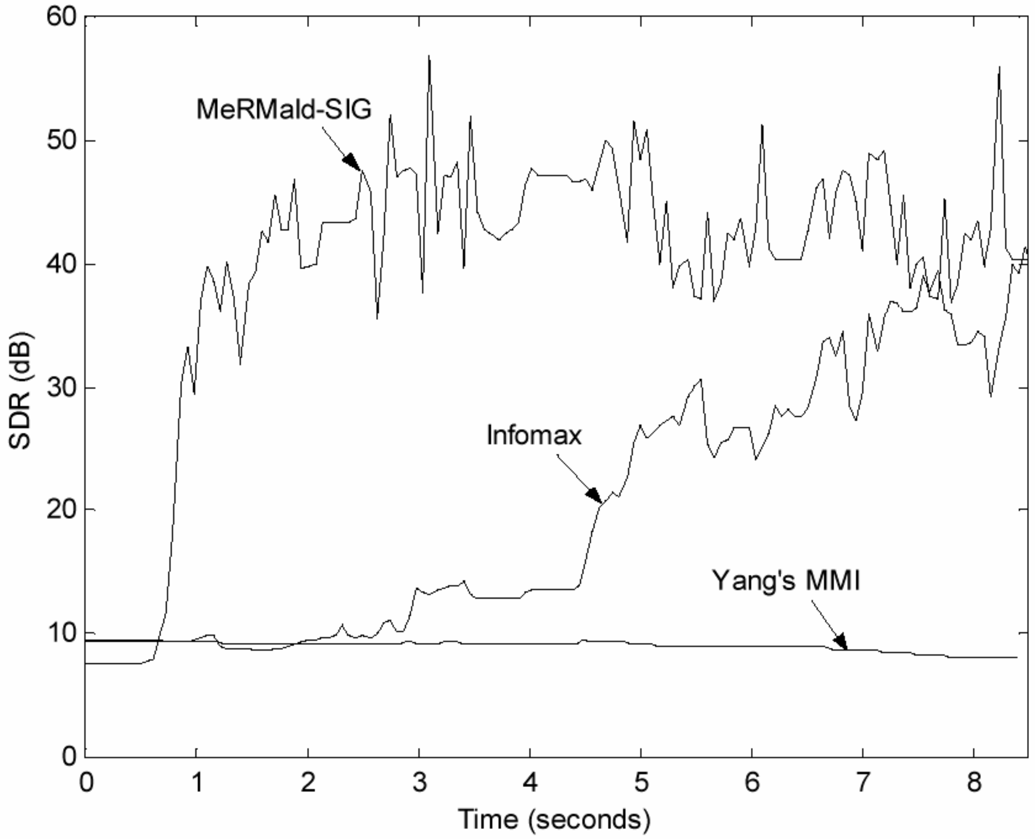


Figure 7-7. Comparison of MeRMaId-SIG with InfoMax and Yang's MMI algorithms in on-line BSS with real recorded audio signals. The SDR values were averaged over 20 Monte Carlo runs using different randomly selected mixing matrices

All three algorithms used the ideal pre-whitening, and as clearly seen, only our SIG-based algorithm was able to converge to the solution fast and accurately. Yang's MMI attained performances as large as 40 dB, but this consistently took much more than the 8.5 seconds shown in the above plot. Larger step sizes for InfoMax and Yang's MMI resulted in instability [Hil01b]. The fast convergence of MeRMaId-SIG in this kind of on-line situations makes it a favorable alternative.

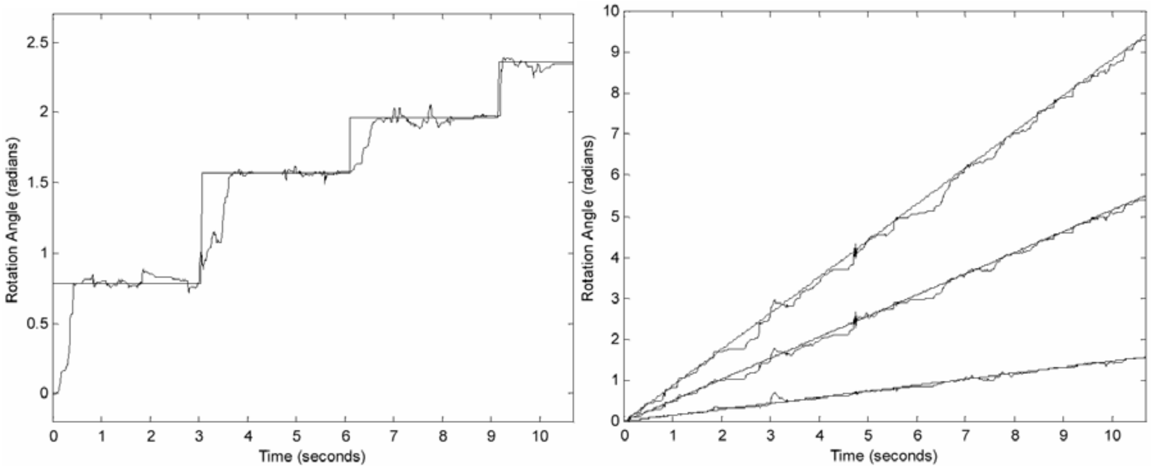


Figure 7-8. Demonstration of the tracking capability of SIG in an on-line BSS situation with time-varying mixture. The ideal solutions and the estimated values are presented for stepwise and linearly varying mixtures

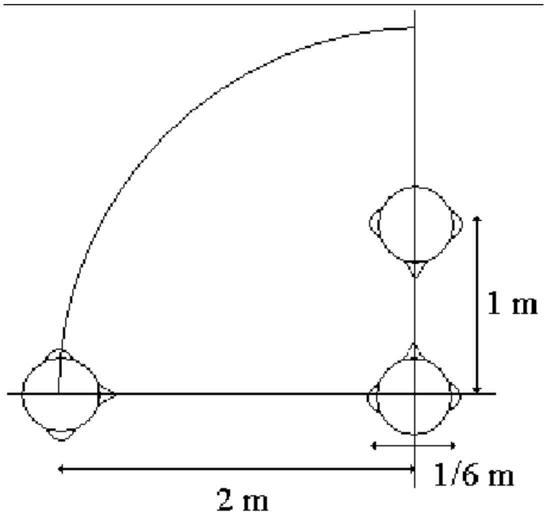


Figure 7-9. Illustration of the geometry for a two-source two-measurement case, where the speakers are the sources and the two ears of the listener are the measurements

As a final example in this case study, we show the tracking capability of MeRMaId-SIG in a time-varying mixture situation. Figure 7-8 shows some results from different scenarios where the rotation portion of the mixture was varied stepwise or linearly at different slopes. The two sources were selected from the 10 audio recordings.

In our last case study, we again perform on-line BSS in a time-varying mixture situation. This time, the variation of the mixture is based on a simplistic spherical wave propagation model for the sound. As simple analogy, consider a listener (with two ears as the sensors that are separated by 1/6m) who has to separate the speeches of two speakers nearby, as shown in Figure 7-9. Our simple model assumes that all speech signals arrive at both sensors simultaneously (so that the mixture is instantaneous) and that the entries of the mixing matrix are determined by the attenuation of the amplitude of the sound waves inversely proportional to the distance traveled by the wave. Under these assumptions, a time-varying mixture occurs when the speakers move around [Hil02].

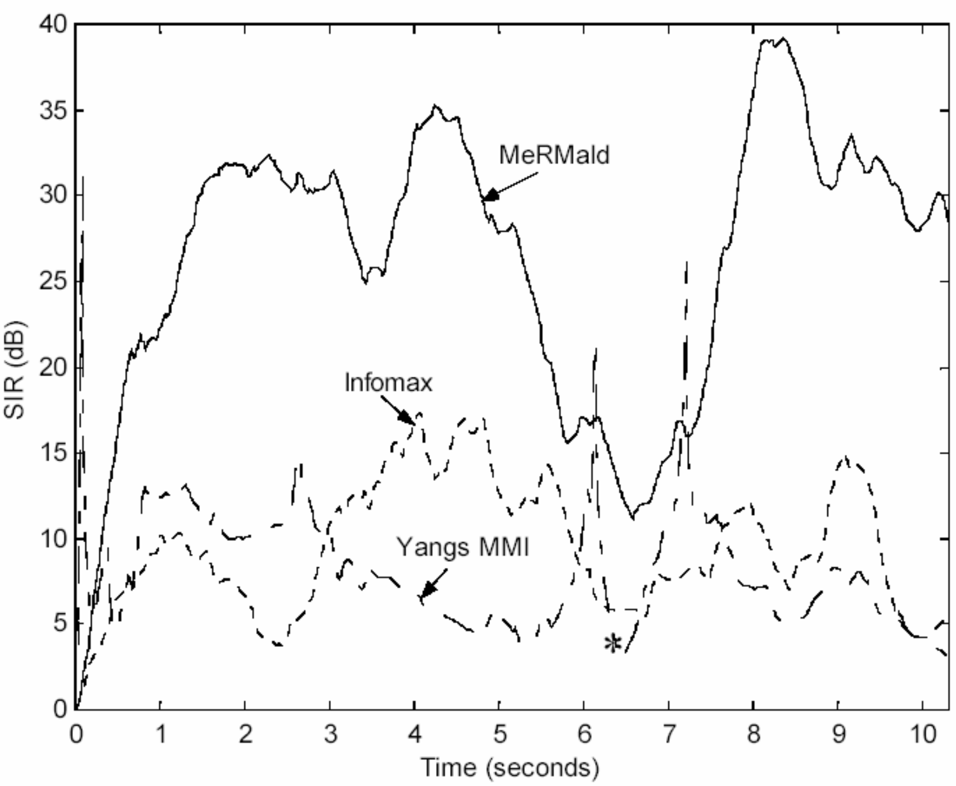


Figure 7-10. The SDR versus time in on-line time-varying BSS using MeRMaId-SIG, InfoMax, and Yang’s MMI, supported by SIPEX-G for pre-whitening. The asterix at 6.3 seconds show the instant where the mixing matrix becomes singular

In these simulations, not only the rotation portion of the mixing matrix is changed (or equivalently we do not assume that the ideal whitening matrix is known). The whitening is also performed on-line with a fast and robust PCA algorithm, which we introduced recently and named as SIPEX-G for *simultaneous principal component extraction using a gradient approach* [Erd02g]. The details of this algorithm are provided in Appendix C. In our first example, we consider a situation where one of the speakers is at a distance of 1m directly in front of the listener and the second speaker is moving counterclockwise at 1m/s on a circle 2m away from the listener, starting from directly in front of the listener. For comparison, we also provide the performance plots for InfoMax and Yang's MMI algorithms, also pre-whitened using the SIPEX-G algorithm. These results are presented in Figure 7-10. Notice that when the second speaker is directly in front of the listener or directly behind, the mixing matrix will become singular, therefore at these positions we expect all algorithms to perform poorly as there is no unique solution to the problem.

In these simulations, the step size for all algorithms were optimized to maximize the SDR performance index and a window length of L=200 samples was used for SIG. Clearly observed from these results, our SIG algorithm achieves a very good solution with over 20 dB signal to distortion ratio in this two-dimensional on-line time-varying BSS problem [Hil02]. In order to test the tracking limits of SIG in this problem, we increase the speed of the second speaker to 5m/s, which will cause 8 points of singularity to occur within the 10.7 seconds shown in the results (we could call these scenarios as the bad-cop good-cop interrogation scenario). The results of MeRMaId-SIG tracking the mixing matrix successfully in this difficult situation is presented in Figure 7-11. As expected, at the points of singularity, the performance of the algorithm degrades, but then recovers quickly once the mixture becomes invertible again [Hil02]. This completes our demonstration of the performance of SIG in on-line adaptation problems. Perhaps, the number of these examples could be extended to many more applications where any type of entropy-based performance index could be used. To sum up, SIG is a very useful tool, whose benefits in the domain of information theoretic learning are equivalent to those of LMS in mean-square learning.

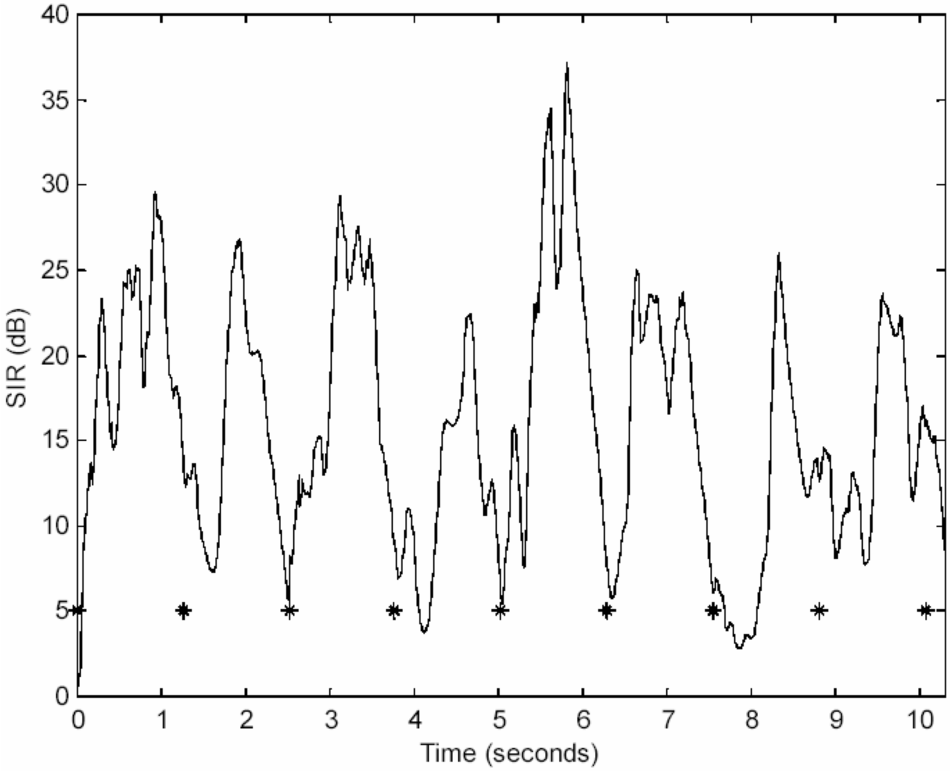


Figure 7-11. The SDR versus time in on-line BSS using MeRMaId-SIG for a very fast changing mixture. SIPEX-G is used for pre-whitening the data on-line

7.4 Extensions to MSE for On-Line Adaptation Based on Observations on SIG

The previous sections of this chapter were devoted to the development and investigation of SIG from an information theoretic learning point-of-view. In this section, we will point out a possible connection between a special case of SIG with second-order criteria in on-line adaptation and based on this observation, we will propose some extensions to the traditional approaches. We start by revisiting the special case SIG expression for an ADALINE structure that corresponded to the choice of Gaussian kernels and L=2, given in Eq. (7.18). This drastically simplified SIG expression is repeated here as Eq. (7.20).

$$\frac{\partial \hat{H}_{\alpha,k}(Y)}{\partial w} = \frac{1}{\sigma^2}(y_k - y_{k-1}) \cdot (x_k - x_{k-1}) \tag{7.20}$$

We will now try to arrive at a similar expression starting from a second-order cost function, not defined on the error, but its time derivative. The traditional MSE criterion in adaptive filtering and on-line learning is given by $E[e^2(t)]/2$. Consider a cost function of the form $E[\dot{e}^2(t)]/2$, where the over-dot represents derivation with respect to time. As always, in order to derive a stochastic gradient for this cost function, we will drop the expectation. Suppose we are to train an ADALINE in a supervised manner using this latter criterion that penalizes large derivatives of the error. Letting $d(t)$ be the desired response and $x(t)$ be the input vector, we get the error signal as $e(t)$=$d(t)$-$w^{\mathrm{T}}x(t)$. Using all these, we can write the stochastic steepest descent algorithm for the weights in continuous time as

$$\dot{w} = -\eta \ \dot{e}(t) \frac{\partial \dot{e}(t)}{\partial w} \tag{7.21}$$

Assuming that the step size is small (therefore $\dot{w}$ is small), we can approximately write $\dot{e}(t) \approx \dot{d}(t) - w^T \dot{x}(t)$, which leads to

$$\dot{w} \approx -\eta \ \dot{e}(t) \frac{\partial \left(\dot{d}(t) - w^T \dot{x}(t)\right)}{\partial w} = \eta \ \dot{e}(t)\dot{x}(t) \tag{7.22}$$

Suppose that we implement this learning algorithm in discrete time using a first order backward difference for the derivatives and a time step of T. Then, the weight update equation for the weights at step k becomes

$$w_{k+1} = w_k + \frac{\eta}{T}(e_k - e_{k-1})(x_k - x_{k-1}) \tag{7.23}$$

This is essentially the same form in Eq. (7.20), modified for the minimization of error entropy (notice that there are two additional sign changes due to minimization of entropy and the definition of error with a negative sign on the weights). Intuitively, we can reason that the derivative of a signal is related to its entropy. For deterministic signals, if we assume that time is a uniform random variable, then the pdf of the signal can be regarded as being uniform (on very short time intervals where the linear approximation is valid) and the entropy of this pdf is proportional to the slope of the signal at the time of consideration. Therefore, it is natural that a cost function defined on the derivative of the signal and minimized using the stochastic gradient, which is a localization in time, gives essentially the same update rule as a cost function defined on the entropy of the signal. These comments, however, should not be taken as rigorous mathematical statements. Although the treatment of deterministic signals as being stochastic for the purposes of evaluating second-order statistical quantities is rather well established in the literature, determining such strong links between information theoretic quantities like entropy and deterministic signals is yet to be developed. These observations can only provide us some insight towards achieving this association. Nevertheless, encouraged by this immediate result, we propose that such relations may be possible in the future.

In on-line system identification of a possibly time-varying dynamical system, it is logical to assume that a performance index that not only targets minimizing the instantaneous value of the squared-error (or any positive function that grows with error values departing from zero), but also considers the time-course of the error signal; in particular, we could talk about a criterion that tries to minimize the instantaneous error while trying to smoothen the deviation of the error from time instant to time instant. In general, to achieve the purpose in continuous time adaptation, we could assume instantaneous performance measures of the form $J(t) = e^2(t) + \lambda_1 \dot{e}^2(t) + \lambda_2 \ddot{e}^2(t) + \ldots$, which are stochastic approximations to $J(t) = E[e^2(t)] + \lambda_1 E[\dot{e}^2(t)] + \lambda_2 E[\ddot{e}^2(t)] + \ldots$. Especially, if a cost function of order one (meaning up to first derivative of error is considered), we could use the derivation above in discrete-time updates of the weight vector of our adaptive system. Specifically for a linear system, these updates would be

$$w_{k+1} = w_k + 2e_k x_k + 2\lambda_1 (e_k - e_{k-1})(x_k - x_{k-1}) \tag{7.24}$$

assuming that all constants are integrated into the variable λ_1. This type of updates will try to decrease the error as long as the *unknown* system stays fixed, and furthermore, they will try to maintain the level of performance should an abrupt change occur in the system that we are trying to model.

In order to achieve faster convergence to the solution of these cost functions, RLS type algorithms could be developed. In fact, while discussing on this possibility, we came up with such an algorithm. The following derivation is due to Yadunandana N. Rao, who is a colleague at CNEL.

Suppose, the cost function $J(w) = E[e_k^2] + \lambda\, E[(e_k - e_{k-1})^2]$ is assumed, where the error samples are generated by the ADALINE system from $e_k = d_k - w^T x_k$. After some algebra, it is possible to show that, this cost function could be equivalently expressed as

$$J = \gamma_d(0) + w^T R w - 2w^T P + \lambda\, (\gamma_{\dot{d}}(0) + w^T R_1 w - 2w^T P_1) \tag{7.25}$$

where we defined

$$\begin{aligned} &R = E[x_k x_k^T] \\ &R_1 = E[(x_k - x_{k-1})(x_k - x_{k-1})^T] \\ &P = E[d_k x_k] \\ &P_1 = E[(d_k - d_{k-1})(x_k - x_{k-1}) \\ &\gamma_d(0) = \text{var}(d_k) \\ &\gamma_{\dot{d}}(0) = \text{var}(d_k - d_{k-1}) \end{aligned} \tag{7.26}$$

Taking the gradient and equating to zero yields the surprising result that the solution is in the same form as the Wiener solution.

$$\begin{aligned} &\frac{\partial J}{\partial w} = 2Rw - 2P + \lambda\ (2R_1 w - 2P_1) = 0 \\ &\Rightarrow w^* = (R + \lambda\ R_1)^{-1}(P + \lambda\ P_1) \end{aligned} \tag{7.27}$$

Letting $Q = (R + \lambda\ R_1)$ and $V = (P + \lambda\ P_1)$, we obtain the following recursions

$$\begin{aligned} Q(k) &= Q(k-1) + (1 + 2\lambda\ )x_k x_k^T - \lambda\ x_{k-1}x_k^T - \lambda\ x_k x_{k-1}^T \\ &= Q(k-1) + 2\lambda\ x_k x_k^T - \lambda\ x_{k-1}x_k^T + x_k x_k^T - \lambda\ x_k x_{k-1}^T \\ &= Q(k-1) + (2\lambda\ x_k - \lambda\ x_{k-1})x_k^T + x_k(x_k - \lambda\ x_{k-1})^T \end{aligned} \tag{7.28}$$

At this point, we make use of the Sherman-Morrison-Woodbury identity, which is

$$(A + BCD^T)^{-1} = A^{-1} - A^{-1}B(C^{-1} + D^T A^{-1}B)^{-1}D^T A^{-1} \tag{7.29}$$

by substituting

$$\begin{aligned} &B = [(2\lambda\ x_k - \lambda\ x_{k-1}) \quad x_k] \\ &D = [x_k \quad (x_k - \lambda\ x_{k-1})] \end{aligned} \tag{7.30}$$

and C=I_{2x2}. With these definitions, we get

$$BCD^T = ... = (2\lambda\ x_k - \lambda\ x_{k-1})x_k^T + x_k(x_k - \lambda\ x_{k-1})^T \tag{7.31}$$

Therefore, the recursion for Q is simply

$$Q(k) = Q(k-1) + BCD^T \tag{7.32}$$

In order to obtain the optimal weights, we need the inverse of $Q(k)$ at each step.

$$Q^{-1}(k) = Q^{-1}(k-1) - Q^{-1}(k-1)B[I_{2x2} + D^T Q^{-1}(k-1)B]^{-1} D^T Q^{-1}(k-1) \quad (7.33)$$

Notice that the inversion is over a 2x2 matrix, which is extremely simple. For $V(k)$, we could simply use the sample mean recursive update. $V(k)$ is explicitly given by (assuming WSS input and desired signals)

$$V(k) = E[(1+2\lambda\ )d_k x_k - \lambda\ d_k x_{k-1} - \lambda\ d_{k-1} x_k] \quad (7.34)$$

The overall complexity of this algorithm is $O(N^2)$, which is the same as the classical RLS algorithm.

Another interesting and promising idea is to use on line supervised adaptation rules that adapt the weight vector on a sample-by-sample basis while maintaining a pre-specified dynamics for the error signal (consider for example a first order dynamics defined by $e_k=\lambda e_{k-1}$). One such update algorithm is derived and discussed in Appendix E.

Although these extensions involving the derivatives of the error signal to the traditional MSE criterion could theoretically be useful in on-line training of adaptive filters under noiseless conditions, how they will behave under noisy situations is a primary issue of interest. Further theoretical and experimental research must be conducted on this issue to determine the applicability of these criteria and to determine any possible gains they might introduce.

CHAPTER 8
RECURSIVE ENTROPY ESTIMATOR AND ITS RECURSIVE GRADIENT

8.1 Derivation of the Recursive Entropy Estimator

In evaluating the statistical properties of a signal on-line, a recursive formula is very useful. For traditional second-order statistical quantities, there are established recursive estimators. Such a recursive equation to estimate entropy on-line and update samples recursively with every incoming sample would be a valuable tool for entropic signal processing.

In this chapter, we will propose two such recursive formulas for our nonparametric quadratic entropy estimator. We will specifically consider the second-order entropy, because an analytically simple derivation with the approach that we will undertake has only been possible for this quantity only. However, we are confident that, in the future, it will be possible to determine recursive estimators for other orders of Renyi's entropy.

In the following derivation, we will concentrate on the quadratic information potential. Once we have a recursive estimate of this quantity, recall that we could easily obtain the corresponding estimate of Renyi's quadratic entropy using the relationship $H_2(X) = -\log V_2(X)$. Consider the quadratic information potential estimator evaluated using (*k*+1) samples of the random variable *X*, which will be denoted by $\hat{V}_{k+1}$ (we will drop the subscript '$_2$' for quadratic from the notation for convenience). In Eq. (8.1), we reorganize terms to get a recursion on its previous value obtained using *k* samples.

$$\begin{aligned}\hat{V}_{k+1} &= \frac{1}{(k+1)^2}\sum_{j=1}^{k+1}\sum_{i=1}^{k+1}\kappa_\sigma(x_j - x_i) \\ &= \frac{1}{(k+1)^2}\left[\begin{array}{l}\sum_{j=1}^{k}\left(\sum_{i=1}^{k}\kappa_\sigma(x_j - x_i) + \kappa_\sigma(x_j - x_{k+1})\right) \\ +\sum_{i=1}^{k}\kappa_\sigma(x_{k+1} - x_i) + \kappa_\sigma(0)\end{array}\right] \\ &= \frac{k^2}{(k+1)^2}\hat{V}_k + \frac{1}{(k+1)^2}\left[2\sum_{i=1}^{k}\kappa_\sigma(x_{k+1} - x_i) + \kappa_\sigma(0)\right]\end{aligned} \tag{8.1}$$

Notice that this is an exact recursion for the nonparametric quadratic information potential estimator, which updates the estimate with each incoming sample, thus it will be called the *exact recursive entropy estimator*. However, it still requires memory to store all previous values. We would rather have a recursion that does not require the storage of all previously obtained samples. To solve this problem, we will have to use a different strategy; instead of trying to recursively estimate the information potential, we will recursively update the pdf estimate. For convenience in comparing the alternative recursive estimator, which will be presented below, with the exact recursion in Eq. (8.1), we introduce a time-varying forgetting factor λ, defined as $\lambda = 1 - k^2/(k+1)^2$ in Eq. (8.1) to obtain the following recursion. Notice that Eq. (8.2) is still an exact recursion for the information potential estimate.

$$\hat{V}_{k+1} = (1-\lambda)\hat{V}_k + \frac{2\lambda}{2k+1}\sum_{i=1}^{k}\kappa_\sigma(x_{k+1} - x_i) + \frac{\lambda}{2k+1}\kappa_\sigma(0) \tag{8.2}$$

As mentioned above, consider now a recursive update of the pdf estimate with every new sample. The initial pdf estimate could be initialized to a kernel centered at the first sample, i.e., $\bar{f}_1(x) = \kappa_\sigma(x - x_1)$.

$$\bar{f}_{k+1}(x) = (1-\lambda)\bar{f}_k(x) + \lambda\kappa_\sigma(x - x_{k+1}) \tag{8.3}$$

Using the definition of information potential, $V_\alpha(X) = E_X[f_X^{\alpha-1}(X)]$, we notice that

$$\begin{aligned} \bar{V}_{k+1}(X) &= E_X[\bar{f}_{k+1}(X)] = E_X[(1-\lambda)\bar{f}_k(X) + \lambda\kappa_\sigma(X - x_{k+1})] \\ &= (1-\lambda)\bar{V}_k(X) + \lambda\ E_X[\kappa_\sigma(X - x_{k+1})] \end{aligned} \tag{8.4}$$

One possibility to approximate the expectation in the last line of Eq. (8.4) is to use the sample mean using a window of samples from the past. Although this would necessitate the storage of these samples, since the window length will be fixed, it is not as big of a problem as it was in the exact recursion case. Using a window length of *L*, the recursive estimator for quadratic information potential with a forgetting factor of λ becomes

$$\bar{V}_{k+1} = (1-\lambda)\bar{V}_k + \frac{\lambda}{L}\sum_{i=k-L+1}^{k}\kappa_\sigma(x_i - x_{k+1}) \tag{8.5}$$

We will call this the *forgetting recursive entropy estimator*. By comparing the two recursions in Eq. (8.2) and Eq. (8.5), we notice that asymptotically they converge to the same value, should the window length *L* in Eq. (8.5) be set to *k* and the forgetting factor to $\lambda = 1 - k^2/(k+1)^2$. To see this result, consider the following limit as the number of samples approach infinity.

$$\lim_{k\to\infty}(\hat{V}_{k+1} - \bar{V}_{k+1}) = \lim_{k\to\infty}\begin{bmatrix} (1-\lambda)\hat{V}_k - (1-\lambda)\bar{V}_k + \frac{\lambda}{2k+1}\kappa_\sigma(0) \\ + \frac{2\lambda}{2k+1}\sum_{i=1}^{k}\kappa_\sigma(x_{k+1} - x_i) - \frac{\lambda}{k}\sum_{i=1}^{k}\kappa_\sigma(x_i - x_{k+1}) \end{bmatrix} = 0 \tag{8.6}$$

The convergence to zero of the terms involving a division by *k* is obvious; for the terms with λ, as *k* approaches infinity, the difference between the two recursions will be multiplied by (1-λ), which is smaller than one for all *k* values (although it approaches to 1 asymptotically). Thus, we conclude that under these special selections of the parameters the two recursions asymptotically converge to the same value. Of course, using a dynamic window length is not something we would do, since the alternative approach used to obtain Eq. (8.5) was used to avoid this in the first place.

These recursive estimates could be used in on-line entropy evaluation of signals for various uses in signal processing; specifically the recursion in Eq. (8.5) could be used to track the entropy of a nonstationary signal and reveal any abrupt pdf changes in the signal (assuming that the pdf change is accompanied by a change in entropy). This, for example, could find applications in time-series segmentation. Perhaps, combined with other statistical properties of the signal, it could also provide a segmentation criterion even for pdfs that share the same entropy value.

8.2 The Recursive Information Gradient

The recursive entropy estimate in Eq. (8.5) is useful for evaluation purposes; however, more important for our purposes of information theoretic learning, its derivative is valuable as it could be used in gradient-based adaptation algorithms. Due to this, we calculate the derivative of Eq. (8.5) with respect to the weight vector of a hypothetical adaptive system that led to the generation of the samples x_k of the random variable X.

Substituting the recursive information potential estimator in Eq. (8.5) into the quadratic Renyi's entropy expression, $\overline{H}_{k+1} = -\log \overline{V}_{k+1}$, we obtain the following recursive gradient for entropy

$$\frac{\partial \overline{H}_{k+1}}{\partial w} = -\frac{\partial \overline{V}_{k+1} / \partial w}{\overline{V}_{k+1}} = -\frac{(1-\lambda)\dfrac{\partial \overline{V}_k}{\partial w} + \dfrac{\lambda}{L}\displaystyle\sum_{i=k-L+1}^{k} \kappa'_\sigma (x_i - x_{k+1}) \left[\frac{\partial x_i}{\partial w} - \frac{\partial x_{k+1}}{\partial w}\right]}{(1-\lambda)\overline{V}_k + \dfrac{\lambda}{L}\displaystyle\sum_{i=k-L+1}^{k} \kappa_\sigma (x_i - x_{k+1})} \quad (8.7)$$

We name this expression the *recursive information gradient* (RIG). Interestingly (it is clear actually if you think about it), SIG in Eq. (7.6) and Eq. (7.11) becomes a special case of RIG corresponding to the specific selection of λ=1. Therefore, we expect RIG to perform the tasks that SIG has completed successfully with even a better performance; explicitly, RIG would provide a smoother estimate of the gradient than SIG, thus result in a smoother convergence to the optimal solution if not faster. Then, of course, there is the intrinsic trade-off in choosing the forgetting factor between tracking capability in nonstationary environments and the final misadjustment after convergence.

One important point in implementing RIG is that the recursion of the gradient is a valid approximation only if the step size used is *sufficiently* small so that the two gradients $\partial \overline{V}_k / \partial w_{k+1}$ and $\partial \overline{V}_k / \partial w_k$ are close to each other. Even though the actual gradient recursion should use the former, which is the gradient with respect to the weights evaluated at the last value of the weight vector, the recursion in Eq. (8.7) uses the latter, which is the gradient from the previous time step.

8.3 Illustration of the Performance of the Recursive Estimators

In this section, we show the accuracy of the recursive entropy estimators and study the effects of the free parameters on various performance aspects of the estimates. First, we start by demonstrating the convergent properties of both estimators to the true entropy value of the pdf underlying the data that is being presented. In these simulations, we used 5000 samples generated by zero-mean, unit-variance uniform, Laplacian, and Gaussian distributions. For these density functions, both the recursion in Eq. (8.1) and the recursion in Eq. (8.5) are evaluated over the samples. The estimated entropy values using a Gaussian kernel with size $\sigma = 0.01$ and the actual entropy of the true pdf of the data are shown in Figure 8-1. For the recursion in Eq. (8.5), the forgetting factor is selected to be 0.005 and the window length is chosen as 100.

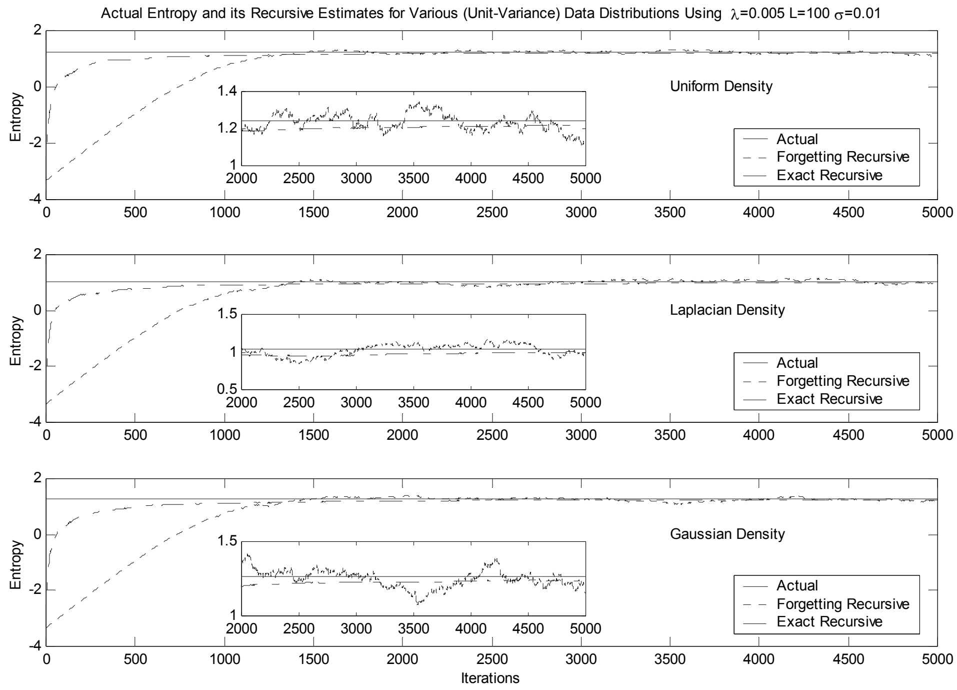


Figure 8-1. Actual entropy and its exact and forgetting recursive estimates for uniform, Laplacian and Gaussian densities

In our second set of simulations, we investigate the effect of the forgetting factor on the convergence time and the convergence accuracy (variance after convergence) of the forgetting estimator in Eq. (8.5). For this purpose, we used this recursion on a uniform density for 10000 iterations. Three different values are used for the forgetting factor: 0.001, 0.003, and 0.01. The convergence plots of the estimates are shown in Figure 8-2. Starting from the same initial estimate, the three recursions converge after approximately 8000, 2500, and 1000 iterations. As expected, the faster the convergence, the larger the estimation variance. When we evaluate the variances of the estimated entropy values over the last 1000 samples of each convergence curve, we see that larger forgetting factors result in larger variance; the variances are respectively, 1.1x10^{-4}, 9.5x10^{-4}, and 2.7x10^{-3}. In these runs, we used L=100 and $\sigma = 0.01$. This result conforms to the well-known general behavior of the forgetting factor in recursive estimates. There is an intrinsic trade-off between speed and variance, which the designer must consider in selecting the forgetting factor.

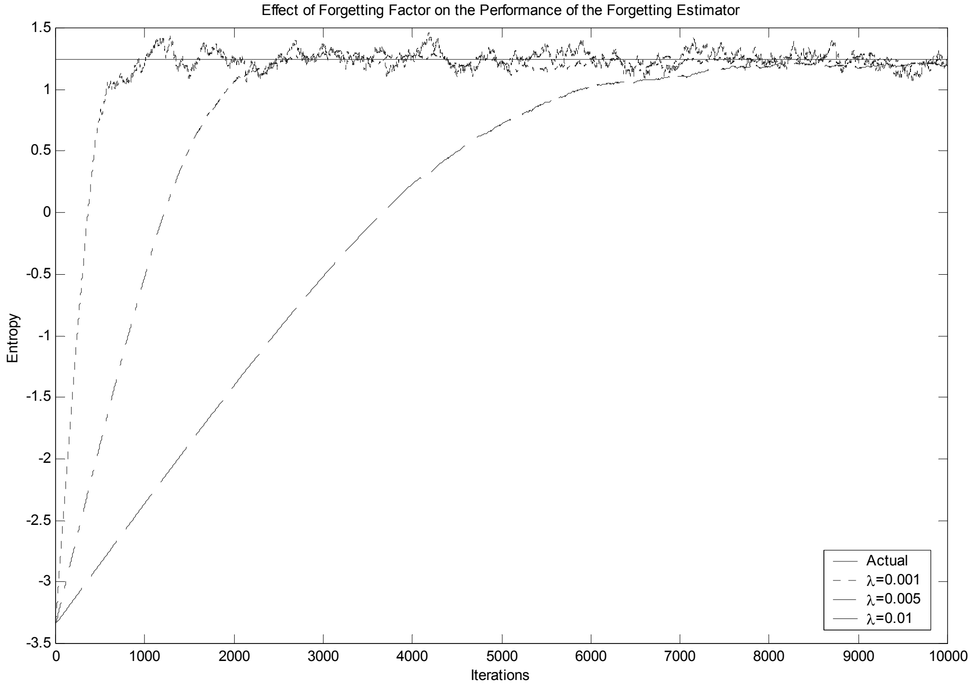


Figure 8-2. Comparison of the convergence properties of the forgetting estimator for different values of the forgetting factor

Our third set of simulations study the effect of the window length that is computed in the sample, which approximates the expectation operator. For this purpose, we fixed the forgetting factor to 0.002, and the kernel size to 0.01 in Eq. (8.5). Three values of L are tried; 10, 100, and 1000. The results of the recursive estimation using these three different window lengths are shown in Figure 8-3. As expected, the speed of convergence is not affected by the variations in this parameter. Only, the estimation variance after convergence is greatly affected. Specifically, the variance of the estimates for these three cases over the last 1000 iterations of the recursion are 6.7x10^{-3}, 7.1x10^{-4}, and 2.2x10^{-5}. This conforms with the general behavior of the sample mean approximation to expectation: The more samples used, the smaller the variance gets.

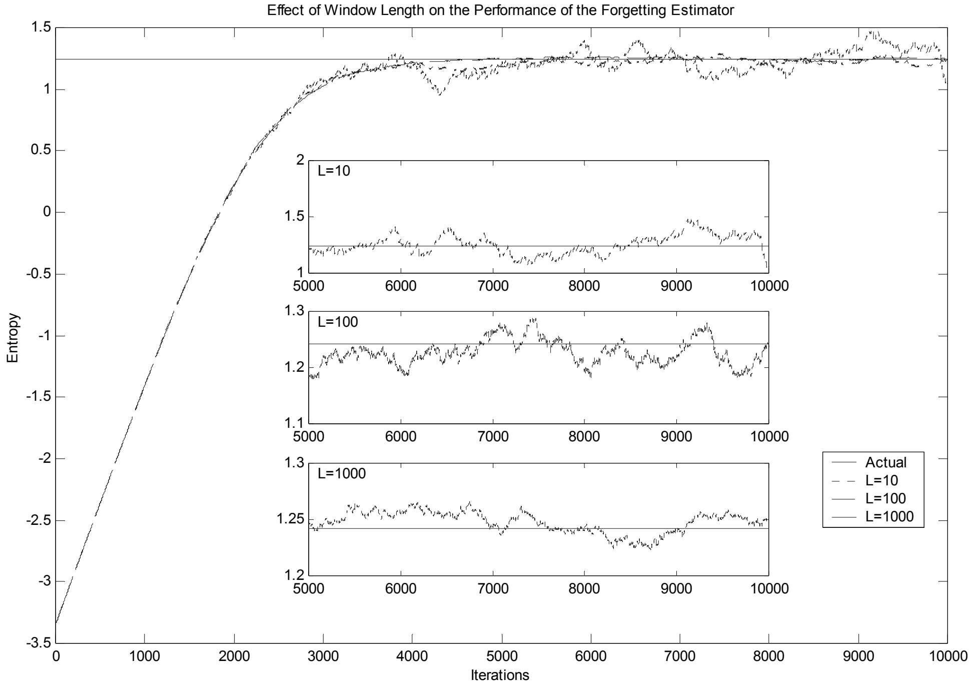


Figure 8-3. Comparison of the convergence properties of the forgetting estimator for different values of the window length

Our fourth set of simulations investigate the effect of kernel size on the variance and bias of the forgetting recursive estimator. As we know, Parzen windowing has a bias that increases with larger kernel sizes, whereas its variance increases with smaller kernel sizes. In accordance with this property of Parzen windowing, we expect our non-parametric estimator to exhibit similar behavior under the variations of kernel size. The convergence plots of the recursions for various values of the kernel size are shown for a uniformly distributed data set in Figure 8-4. In all runs, the forgetting factor was fixed to 0.002 and the window length was taken as 100. For the kernel size values of 0.001, 0.01, 0.1, and 1, the bias over the last 1000 samples of the recursion turned out to be 5.1x10^{-2}, 2.2x10^{-2}, 1.3x10^{-2}, and 2.4x10^{-1}; the variances were also computed and found to be 3.9x10^{-3}, 1.6x10^{-4}, 2.9x10^{-5}, and 3.4x10^{-5}. As expected, the smallest kernel size resulted in the largest variance and the largest kernel size resulted in the largest bias.

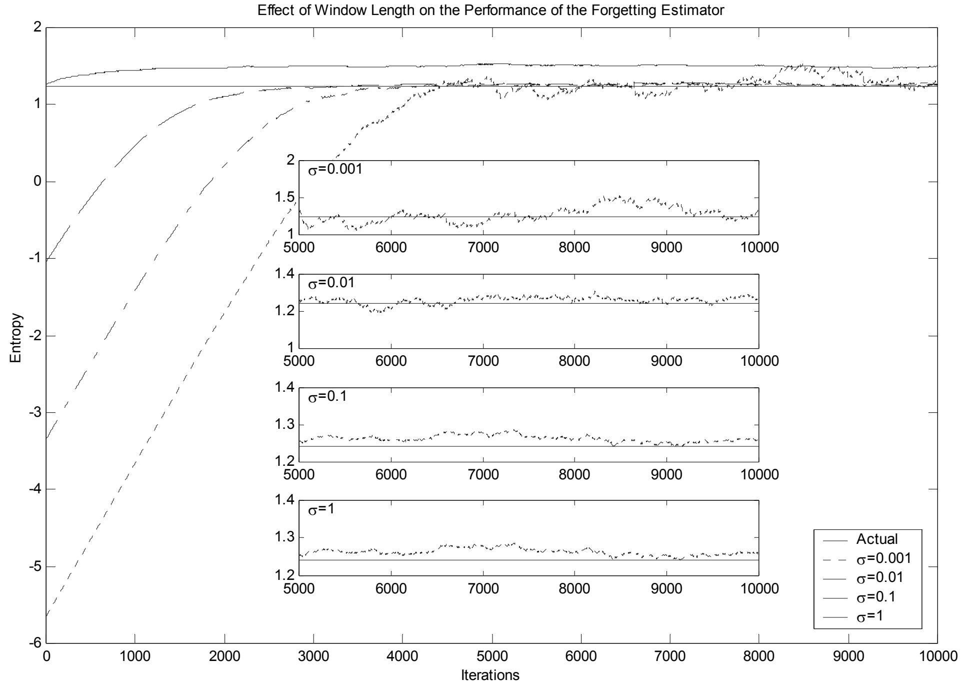


Figure 8-4. Comparison of the convergence properties of the forgetting estimator for different values of the kernel size

Our fifth simulation shows the tracking capability of the forgetting estimator in Eq. (8.5). For this simulation, we used a forgetting factor of 0.002, a window length of 100, and a base kernel size of 0.01. The recursion is initialized to the entropy of the kernel function. In order to enhance the differences between the entropies of the uniform, Laplacian, and Gaussian pdfs, we scaled their standard deviations by the coefficients 1, 5, and 0.2 respectively. At the switching instant, the kernel size of the estimator is also scaled up or down from the base kernel size given above at the same ratio with the standard deviation. Although in a practical situation, we would not know the instant of switching between pdfs and scales, we could still predict the standard deviation of the samples with a forgetting recursion and use this value as a measure for modifying the kernel size. This procedure, however, is beyond the scope of our discussion in this example, therefore, we make use of the true scale factors.

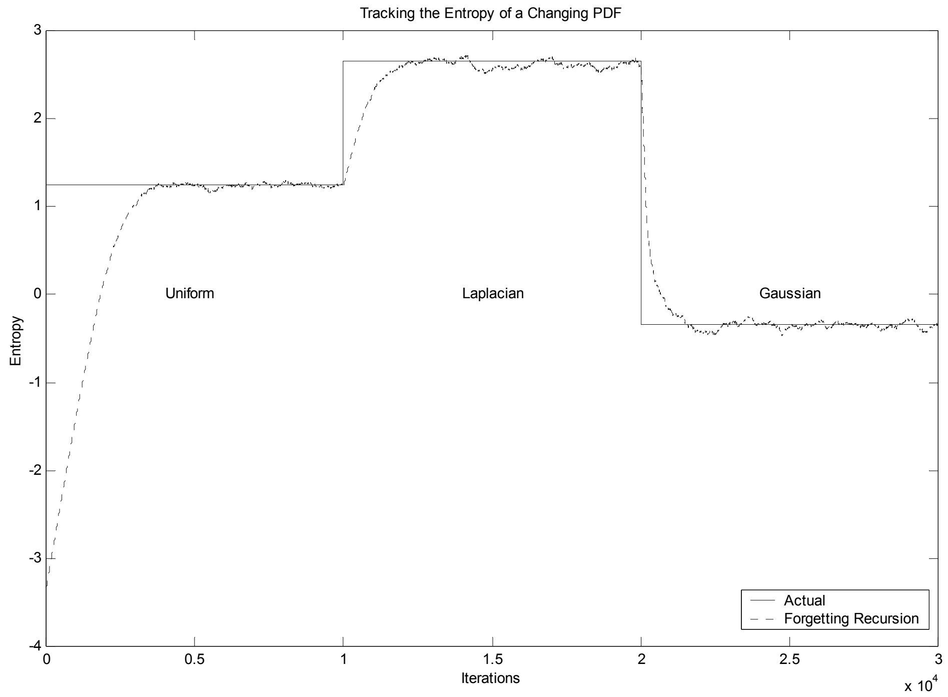


Figure 8-5. Tracking the entropy of a signal with (step-wise) time-varying pdf

In order to address the issue of performance under the situation where the actual value of the scale factor is unknown, we performed an additional simulation using the proposed approach and the recursive sample variance estimator

$$\operatorname{var}(x)_{k+1} = (1-\lambda)\operatorname{var}(x)_k + \lambda\ x_k^2 \tag{8.8}$$

assuming the same forgetting factor value of 0.002 for both the variance and the entropy recursions. The base kernel size is set to 0.01 and the window length is again 100. The algorithm is presented with a sequence of 30000 random samples generated by zero-mean uniform, Laplacian, and Gaussian distributions with standard deviations 10, 1, and 30 respectively. The initial scale factor estimate (i.e., the estimate of the standard deviation of the pdf underlying the samples) is set to 1. We observe from Figure 8-6 that even though the scale estimates are not accurate, the entropy estimates converge towards the actual entropy value and as soon as the scale factor estimate converges, the difference between the two entropy estimates that use the estimated and actual values of the scale factors drop back to zero.

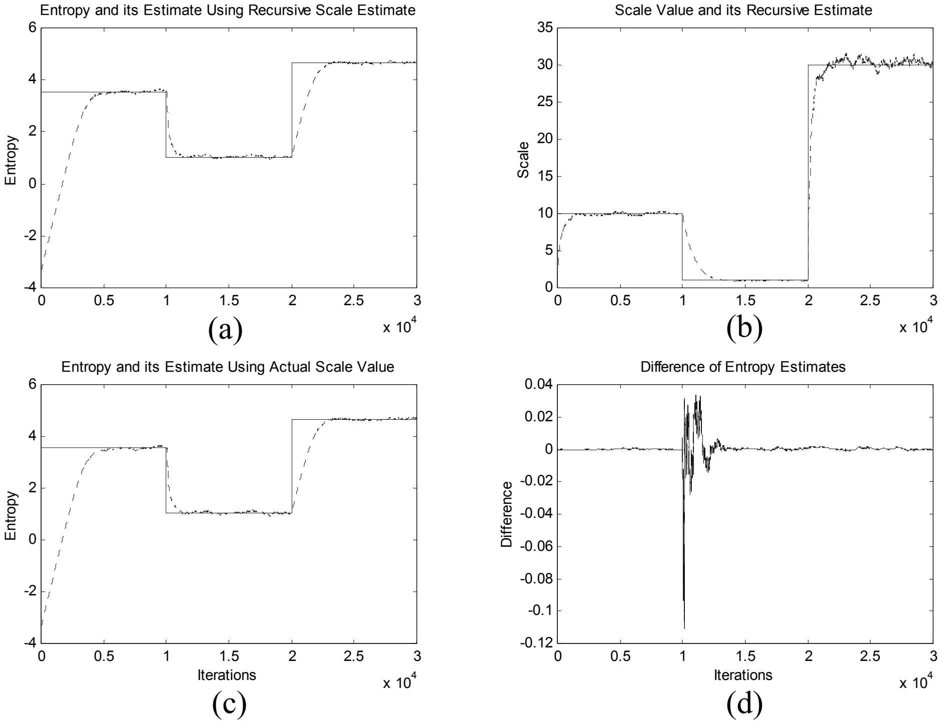


Figure 8-6. Comparison of entropy estimates using the actual and estimated values of the scale factor a) entropy estimate using the estimated scale factor b) scale factor estimate c) entropy estimate using the actual scale factor d) difference between the two entropy estimates using the actual and estimated values of the scale factor

In this chapter, we introduced two recursive estimators for entropy that produce updated estimates after processing each new sample. One of these estimators is an exact recursive formulation of the batch mode quadratic entropy estimator, suitable for stationary signals, and the second one uses a forgetting factor, which makes it suitable for nonstationary signals and tracking changes in entropy. We studied the performance of these estimators in terms of convergence speed and accuracy, analyzing the effects of the free design parameters like the forgetting factor, window length and kernel size on the speed and final estimation variance and bias. Simulations showed the usefulness of these recursive estimators for on-line entropy estimation in various applications of signal processing. In addition, we calculated the gradient of the *forgetting recursive estimator* and showed that SIG is a special case of this recursive entropy gradient, which we called RIG, corresponding to a forgetting factor of 1.

CHAPTER 9
EXTENSION TO FANO'S BOUND USING RENYI'S ENTROPY

9.1 Introduction

Fano's bound is a well-known inequality in the information theory literature [Fan61]. It is essential to the proofs of key theorems [Cov91]. Applied to a classifier, by providing a lower bound for classification error probability, it is useful in terms of giving an indication of attainable performance. In addition, it provides some insights as to how the process of information transfer progresses in this setting, linking classification performance with information theory. In fact, this is one of the outstanding advantages of information theory, the abstract level of investigation and analysis. Linsker's infomax principle progresses along similar lines. As a principle for self-organization, infomax states that an optimal system must transfer as much information as possible from its input to its output, i.e., maximize the mutual information between its input and output [Lin88]. Fano's bound entails similar conclusions about the structure of optimal classifiers; these must maximize the mutual information between actual and decision classes to minimize the probability of error [Tor00].

The question of determining optimal features has been one of the major focal points in pattern recognition research, and information theory has played a central role in this quest [Fu70, Fuk72]. It has been established that information is not preserved in subspace projections, yet maximization of information across the mapping is essential in this process [Dec96]. Fisher and Torkkola recently used this approach to train neural networks directly from samples for optimal feature extraction using the nonparametric estimator for the quadratic Renyi's entropy [Fis97, Tor00]. In all of these, Fano's bound appears as the central-piece because it relates classification error to conditional entropy. Although Fano's lower bound for the probability of error in classification is a valuable indicator of attainable performance, the goal in statistical pattern recognition and machine learning is to minimize the probability of error [Rip96], or possibly an upper bound for the error probability as in structural risk minimization [Vap95]. Therefore, a family of lower and upper bounds would encompass the advantages of both; identify the limitations and indicate the possible generalization performance simultaneously.

Fano's inequality is derived utilizing Shannon's entropy definition [Fan61]. Motivated by Shannon's brilliant work [Sha48, Sha64], researchers concentrated their efforts on information theory. Renyi was able to also formulate the theory of information starting from four basic postulates [Ren70]. His definitions of information theoretic quantities like entropy and mutual information encompassed Shannon's definitions as special cases. Inspired by Fano's bound, many researchers also proposed modifications, generalizations, or alternative information theoretic inequalities, mainly with applications to communication theory [Bas78, Fed94, Gal68, Han94, Poo95]. The recent work of Feder and Merhav is especially important as it provides a lower and an upper bound for the minimal probability of error in estimating the value of a discrete random variable [Fed94]. Their bounds show the association between the probability of value-prediction error and Shannon's entropy, and Renyi's entropy of order infinity as well. Han and Verdu's generalization to Fano's bound, again using Renyi's entropy of order infinity is theoretically appealing and also useful in proving a generalized source-channel separation theorem [Han94]. Yet, the bounds presented in these works do not explicitly consider the classification process, thus do not make use of the confusion matrix of the classifier under consideration. Nevertheless, motivated by these works that extend on classical results utilizing Renyi's alternative definition of information, we developed a family of lower and upper bounds, using Renyi's definitions of information theoretic quantities. For this, the free parameter in Renyi's definitions was exploited along with Jensen's inequality for convex and concave functions.

9.2 Entropy and Mutual Information for Discrete Random Variables

In the development of the aforementioned bounds, we will use several information theoretic quantities as defined by Shannon and Renyi. These are the joint entropy, (average) conditional entropy, and (average) mutual information. We use the random variable *M* to denote the actual class (input space) and *W* to denote the decided class (output space) when applying these arguments to classifiers with a known confusion matrix and priors. The random variable *E*, which takes the values *e* or *c*, is used to denote the events of wrong and correct classification with probabilities $\{p_e,1\text{-}p_\mathrm{e}\}$

9.2.1 Shannon's Definitions

For a discrete random variable *M*, whose probability mass function (pmf) is $\{p(m_k)\}_{k=1}^{N_c}$, Shannon's entropy is given by [Sha48]

$$H_S(M) = -\sum_{k=1}^{N_c} p(m_k)\log p(m_k) \tag{9.1}$$

Based on this definition, the joint entropy, mutual information, and conditional entropy are defined as

$$
\begin{aligned}
H_S(M,W) &= -\sum_{k=1}^{N_c}\sum_{j=1}^{N_c} p(m_k,w_j)\log p(m_k,w_j) \\
I_S(M,W) &= \sum_{k=1}^{N_c}\sum_{j=1}^{N_c} p(m_k,w_j)\log\frac{p(m_k,w_j)}{p(m_k)p(w_j)} \\
H_S(M\,|\,W) &= \sum_{k=1}^{N_c} H_S(W\,|\,m_k)p(m_k)
\end{aligned}
\tag{9.2}
$$

where

$$H_S(W\,|\,m_k) = -\sum_{j=1}^{N_c} p(w_j\,|\,m_k)\log p(w_j\,|\,m_k) \tag{9.3}$$

and $p(m_k,w_j)$ and $p(m_k|w_j)$ are respectively the joint probability mass function and the conditional probability mass function of M given W, respectively. Shannon's mutual information is equal to the Kullback-Leibler divergence [Kul68] between the joint distribution and the product of marginal distributions, and it satisfies the following property [Fan61].

$$I_S(M,W) = H_S(W) - H_S(W\,|\,M) \tag{9.4}$$

9.2.2 Renyi's Definitions

Renyi's entropy for M is given by [Ren70]

$$H_\alpha(M) = \frac{1}{1-\alpha}\log\sum_{k=1}^{N_c} p^\alpha(m_k) \tag{9.5}$$

where α is a real positive constant different from 1, as in the continuous random variable definition. The (average) mutual information and (average) conditional entropy formulations are consequently found to be as given in Eq. (9.6). These definitions are based on the original entropy definition and derived using basic postulates about information.

$$H_\alpha(M,W) = \frac{1}{1-\alpha}\log\sum_{k=1}^{N_c}\sum_{j=1}^{N_c} p^\alpha(m_k, w_j)$$

$$I_\alpha(M,W) = \frac{1}{\alpha-1}\log\sum_{k=1}^{N_c}\sum_{j=1}^{N_c} \frac{p^\alpha(m_k, w_j)}{p^{\alpha-1}(m_k)p^{\alpha-1}(w_j)} \tag{9.6}$$

$$H_\alpha(W \mid M) = \sum_{k=1}^{N_c} p(m_k) H_\alpha(W \mid m_k)$$

where

$$H_\alpha(W \mid m_k) = \frac{1}{1-\alpha}\log\sum_{j=1}^{N_c} p^\alpha(w_j \mid m_k) \tag{9.7}$$

The entropy order α in Renyi's definitions will be helpful in the following sections, when we apply Jensen's inequality to obtain the lower and upper bounds for the probability of error. In order to perceive the effect of α on the value of entropy, consider the following fact: Renyi's entropy is a monotonically decreasing function of α whose values range from $\log N_c$ to $-\log(\max_k p(m_k))$ as it is varied from zero to infinity. It could be shown easily using L'Hopital's rule that the limit of Renyi's entropy (and mutual information) for discrete random variables approach Shannon's definitions as α goes to 1.

9.3 Fano's Bound on Misclassification Probability

Fano's inequality determines a lower bound for the probability of classification error in terms of the information transferred through the classifier. More specifically, consider a classifier for which the actual classes, denoted by M, have prior probabilities $\{p(m_k)\}_{k=1}^{N_c}$ and the decided classes, denoted by W, have the conditional probabilities $p(w_j| m_k)$. Fano's bound for the probability of classification error, in accordance with the definitions of the previous section and in terms of the conditional entropy, is then given by [Fan61]

$$p_e \geq \frac{H_S(W \mid M) - h_S(p_e)}{\log(N_c - 1)} \tag{9.8}$$

where the special notation $h_S(p_e) = -p_e \log p_e - (1-p_e)\log(1-p_e)$ is used for binary Shannon's entropy. Notice that this original bound, as it appears in Fano's derivation has the probability, has the probability of error appearing on both sides of the inequality. Also the denominator prevents the application of this bound to two-class situations. To account for these problems, the binary entropy of p_e is replaced by its maximum possible value, $\log_2 2=1$, and the denominator is replaced with the larger $\log N_c$. In addition, the conditional entropy is replaced by the sum of marginal entropy and mutual information terms in accordance with Eq. (9.4). After all these modifications, the commonly presented version of Fano's bound in the literature is [Tor00]

$$p_e \geq \frac{H_S(W) - I_S(M;W) - 1}{\log N_c} \tag{9.9}$$

9.4 Bounds Using Renyi's Entropy and Mutual Information

We applied Jensen's inequality on Renyi's definition of conditional entropy, joint entropy and mutual information to obtain the following lower and upper bounds for the probability of error [Erd01a, Erd02e, Erd02f]. Since Renyi's mutual information and conditional entropy do not share the identity in Eq. (9.4), these bounds had to be separately derived, starting from their corresponding basic definitions. For convenience, we provide the derivation for the bound that uses the conditional entropy below. The derivations of the bounds using the joint entropy and the mutual information are given in Appendix D. In this derivation, we will use the well-known Jensen's inequality, which has found application in the derivation of many theoretically useful bounds. This inequality is described below for convenience.

Jensen's Inequality: Assume that $g(x)$ is a convex function (if concave reverse inequality), and $x \in [a,b]$, then for $\sum_k w_k = 1,\ w_k > 0$, we have the inequality $g\left(\sum_k w_k x_k\right) \le \sum_k w_k g(x_k)$.

For later use in the derivation, we also write the conditional probability of error given a specific input class as

$$\begin{aligned} p(e \mid m_k) &= \sum_{j \neq k} p(w_j \mid m_k) \\ 1 - p(e \mid m_k) &= p(w_k \mid m_k) \end{aligned} \tag{9.10}$$

Consider Renyi's conditional entropy of *W* given *m_k*.

$$\begin{aligned} H_\alpha(W \mid m_k) &= \frac{1}{1-\alpha} \log \sum_j p^\alpha(w_j \mid m_k) \\ &= \frac{1}{1-\alpha} \log \left[\sum_{j \neq k} p^\alpha(w_j \mid m_k) + p^\alpha(w_k \mid m_k) \right] \\ &= \frac{1}{1-\alpha} \log \left[p^\alpha(e \mid m_k) \sum_{j \neq k} \left(\frac{p(w_j \mid m_k)}{p(e \mid m_k)} \right)^\alpha + (1 - p(e \mid m_k))^\alpha \right] \end{aligned} \tag{9.11}$$

Using Jensen's inequality, and Eq. (9.10), we obtain two inequalities for $\alpha > 1$ and $\alpha < 1$ cases.

$$\begin{aligned} H_\alpha(W \mid m_k) &\underset{\alpha<1}{\overset{\alpha>1}{\substack{\le \\ \ge}}} p(e \mid m_k) \frac{1}{1-\alpha} \log p^{\alpha-1}(e \mid m_k) \sum_{j \neq k} \left(\frac{p(w_j \mid m_k)}{p(e \mid m_k)} \right)^\alpha \\ &\qquad + (1 - p(e \mid m_k)) \frac{1}{1-\alpha} \log (1 - p(e \mid m_k))^{\alpha-1} \\ &= H_S(e \mid m_k) + p(e \mid m_k) \frac{1}{1-\alpha} \log \sum_{j \neq k} \left(\frac{p(w_j \mid m_k)}{p(e \mid m_k)} \right)^\alpha \end{aligned} \tag{9.12}$$

Recall that for an (N_c-*1*)-point entropy we have the following upper bound, which is the entropy of a uniform probability distribution.

$$\frac{1}{1-\alpha}\log\sum_{j\neq k}\left(\frac{p(w_j \mid m_k)}{p(e\mid m_k)}\right)^{\alpha} \leq \log(N_c - 1) \tag{9.13}$$

equality being achieved only for a uniform distribution. Hence, for $\alpha > 1$, from Eq. (9.12) and Eq. (9.13) we obtain

$$H_{\alpha}(W \mid m_k) \leq H_S(e \mid m_k) + p(e\mid m_k)\log(N_c - 1) \tag{9.14}$$

Finally, using Baye's rule on the conditional distributions and entropies we get the lower bound for p_e.

$$H_{\alpha}(W \mid M) \leq H_S(e) + p_e \log(N_c - 1) \tag{9.15}$$

For $\alpha < 1$, from Eq. (9.12) we have

$$\begin{aligned} H_{\alpha}(W \mid m_k) &\geq H_S(e\mid m_k) + p(e\mid m_k) H_{\alpha}(W \mid e, m_k) \\ &\geq H_S(e\mid m_k) + p(e\mid m_k)[\min_k H_{\alpha}(W \mid e, m_k)] \end{aligned} \tag{9.16}$$

where the ‘conditional entropy given an error is made in classification and actual class was m_k’ is

$$H_{\alpha}(W \mid e, m_k) = \frac{1}{1-\alpha}\log\sum_{j\neq k}\left(\frac{p(w_j \mid m_k)}{p(e\mid m_k)}\right)^{\alpha} \tag{9.17}$$

Finally, combining these results and fusing Fano’s special case into the lower bound, we obtain the following interval for classification error probability.

$$L = \frac{H_{\alpha}(W\mid M) - h_S(p_e)}{\log(N_c - 1)} \leq p_e \leq \frac{H_{\beta}(W\mid M) - h_S(p_e)}{\min_k H_{\beta}(W \mid e, m_k)} = U, \quad \begin{matrix} \alpha \geq 1 \\ \beta < 1 \end{matrix} \tag{9.18}$$

Following a similar approach (described in Appendix F), we obtain the following bounds expressed in terms of the joint entropy and the mutual information.

$$\frac{H_{\alpha}(W,M) - H_S(M) - h_S(p_e)}{\log(N_c - 1)} \leq p_e \leq \frac{H_{\beta}(W,M) - H_S(M) - h_S(p_e)}{\min_k H_{\beta}(W \mid e, m_k)}, \quad \begin{matrix} \alpha \geq 1 \\ \beta < 1 \end{matrix} \tag{9.19}$$

$$\frac{H_S(W) - I_\alpha(W;M) - h_S(p_e)}{\log(N_c - 1)} \le p_e \le \frac{H_S(W) - I_\beta(W;M) - h_S(p_e)}{\min\limits_k H_S(W \mid e, m_k)}, \quad \begin{array}{l} \alpha \ge 1 \\ \beta < 1 \end{array} \qquad (9.20)$$

Notice that in all three cases, the lower bounds for α =1 corresponds to Fano's bound through equality Eq. (9.4). The term in the denominator of the upper bound is the entropy of the conditional distribution given the actual class and that the classifier makes an error.

From a theoretical point of view, these bounds are interesting as they indicate how the information transfer through the classifier relates to its performance. Since the family parameter of Renyi's definition does not affect the location of minimum and maximum points of the entropy and mutual information, it is safely concluded, for example, from Eq. (9.20) that, as the mutual information between the input and the output of the classifier is increased its probability of error decreases. Consequently, this result also provides a theoretical basis for utilizing mutual information for feature extraction.

The denominators of the upper bounds also offer an interesting insight about the success of the classification process. As these entropy terms are maximized, the upper bounds become tighter. This happens when the corresponding distribution is uniform; that is when the distribution of probabilities over the erroneous classes is uniform. This conclusion conforms the observations of Feder and Merhav [Fed94]. They also noted that in a prediction process, their upper bound is tightest when the probabilities are distributed uniformly over the *wrong* values.

Recall that Renyi's entropy is a monotonously decreasing function of the entropy order. Therefore, it is clear that the lower bound in Eq. (9.18) attains its tightest (i.e., greatest) value for Shannon's entropy, which is exactly the Fano's bound. Determining the tightest upper bound is not as easy. The optimal value of the entropy order is determined by the balance between the decrease in the numerator and the increase in the denominator. However, our simulations with several simple examples point out that the tightest value for the upper bounds may as well be attained for values of entropy order approaching to 1. These simulation results will be presented below.

One issue to be solved in these bound expressions (also an issue for the original bound by Fano) is to eliminate the binary entropy of the probability of error from the bounds; otherwise, the probability of error appears in both sides of the inequalities. For theoretical use, this may not cause any problems (as evident from the wide use of Fano's bound in various proofs in information theory). From a practical point of view, however, this situation must be corrected. We investigated ways of achieving this objective [Erd02e, Erd02f], however, the obtained bounds were extremely loose compared to the original bounds. Therefore, we do not present these modified bounds here. We could use the bounds as they appear in Eqs. (9.18)-(9.20) by nonparametrically estimating the confusion matrix and the prior probabilities (perhaps by simply counting samples). On the other hand, the information used in this approach is already sufficient to estimate directly the probability of error itself. Therefore, we suggest using the estimated bounds as a confirmation of the estimated probability of error. They may also provide confidence intervals on the calculated value. For a practical application of this procedure, however, further work and analysis of the bounds estimated from a finite number of samples is necessary [Erd02f].

9.5 Numerical Illustrations of the Bounds

In this section, we show the performance of the bounds in a number of different numerical case studies. These studies are aimed to show the basic conclusions drawn in the preceding sections about these extended Fano's bounds. In addition, we will present a comparison of these bounds and the Feder & Merhav bound applied to misclassification probability through one of the examples.

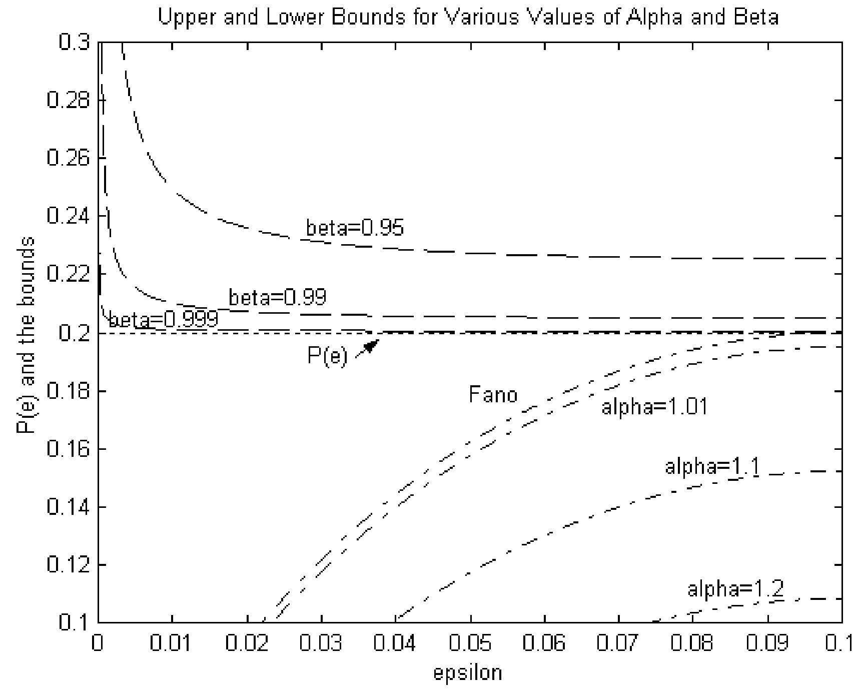


Figure 9-1. Family of lower and upper bounds for probability of error evaluated for different values of the free parameter

Our first example is a simple 3-class situation designed to test the basic properties of the bounds. For this example, the confusion matrix of our *hypothetical* classifier is given by

$$P_{W|M} = \begin{bmatrix} 1-p_e & p_e-\varepsilon & \varepsilon \\ \varepsilon & 1-p_e & p_e-\varepsilon \\ p_e-\varepsilon & \varepsilon & 1-p_e \end{bmatrix} \tag{9.21}$$

whose ij^{th} entry denotes the conditional probability of decision on class-*i* given the input class-*j*. Each column represents the distribution of the probabilities among the possible output classes and the diagonal entries correspond to the probabilities of correct classification given a specific input class. The structure of this confusion matrix guarantees that the overall probability of error is fixed at p_e, which is selected to be 0.2 in the following examples. By varying the free variable ε in the interval $[0, p_e/2]$, it is possible to study the performance of the bounds in terms of tightness. The lower and upper bounds of Eq. (9.18) evaluated for various values of the family parameter (entropy order) are shown in Figure 9-1 as a function of ε. We observe that the family of lower bounds achieves its tightest value for Fano's bound, whereas the upper bounds become tighter as the entropy order approaches one. One other interesting observation is that, the upper bounds remain virtually flat over a wide range of ε suggesting that this bound is as tight for a broad variety of classifiers as it is for the optimum situation where the probability mass distribution among the *wrong* output classes is uniform. If the upper bound is evaluated for β=1, then it reduces to exactly the probability of error, p_e.

In the same setting, we compare the three different versions of the bounds given in Eqs. (9.18)-(9.20). This time, however, we use two sets of prior probabilities to see the effect of this change on the bounds. The results shown in Figure 9-2 clearly show that the bounds that use the conditional entropy and the mutual information are not susceptible to changes in class priors, whereas in the case of uniform priors the bound using the joint entropy also achieves the same level of performance with the other two.

For a comparison with the Feder & Merhav bound, we use the same example once again. The upper bound uses version Eq. (9.18) with entropy order 0.995, where as for the lower bound, Fano's bound is used. The class priors are selected to be uniform. Figure 9-3 depicts these bounds as a function of ε for three different values of the probability of error. Notice that the relative performances of the bounds remain constant. The plot also includes our (loose) modified upper bound [Erd02f].

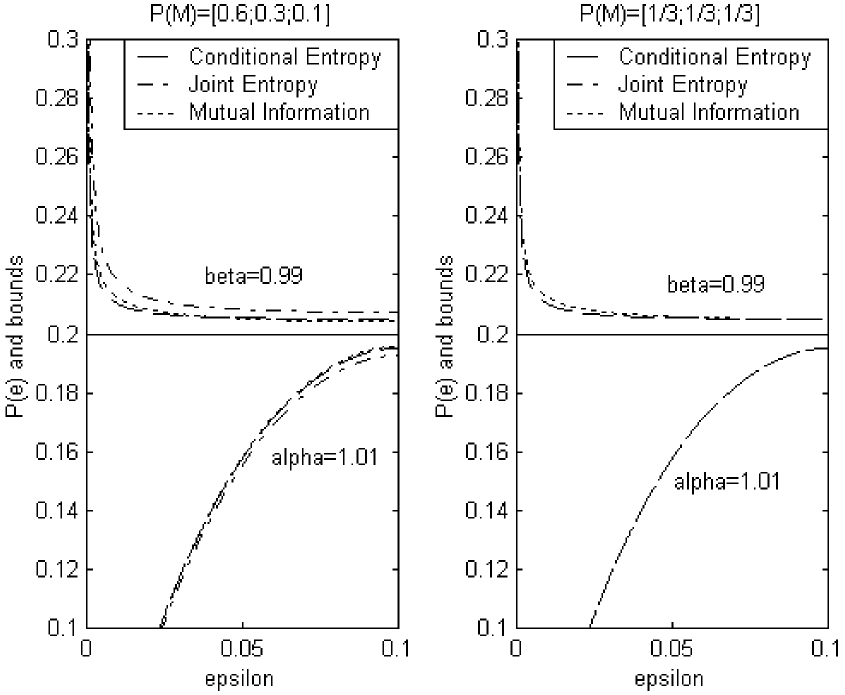


Figure 9-2. Bounds evaluated using different versions (conditional entropy, joint entropy, and mutual information) for different choices of the prior probabilities

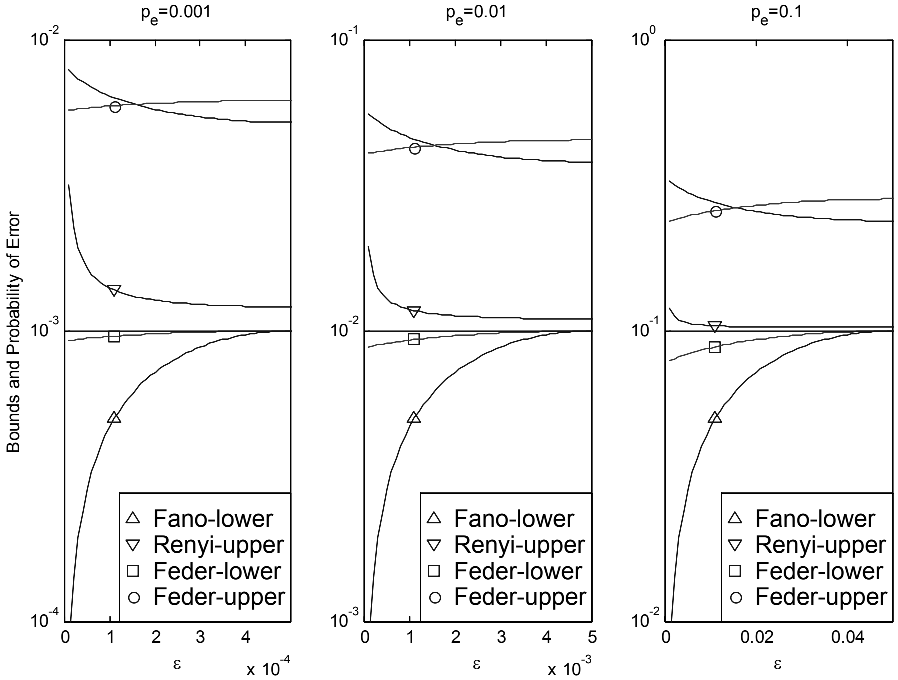


Figure 9-3. Probability of error – constant with respect to ε (dotted), original Fano's lower (Δ) and Renyi's upper (∇) bounds, Feder & Merhav's upper (o) and lower (□) bounds, and the modified Renyi's upper bound (dotted) vs ε

As a second example, we evaluate the bounds for an oversimplified QPSK digital communication scheme over an AWGN channel. The energy per transmitted bit is E_b and the PSD for the additive white Gaussian noise is $N_0/2$. In this problem, it is possible to evaluate the exact values for average bit error rate p_e and all the conditional and prior probabilities necessary to evaluate the bounds in terms of Q-functions. The confusion matrix for this case is

$$P_{W|M}^{QPSK} = \begin{bmatrix} (1-Q_1)^2 & Q_1*(1-Q_1) & Q_1^2 & Q_1*(1-Q_1) \\ Q_1*(1-Q_1) & (1-Q_1)^2 & Q_1*(1-Q_1) & Q_1^2 \\ Q_1*(1-Q_1) & Q_1*(1-Q_1) & (1-Q_1)^2 & Q_1*(1-Q_1) \\ Q_1^2 & Q_1^2 & Q_1*(1-Q_1) & (1-Q_1)^2 \end{bmatrix} \tag{9.22}$$

where $Q_1 = Q\left(\sqrt{2E_b / N_0}\right)$. The prior probabilities for each symbol are assumed to be uniformly distributed. The probability of error and the bounds are shown in Figure 9-4. For the upper bound, we used entropy order 0.995 in Eq. (9.18).

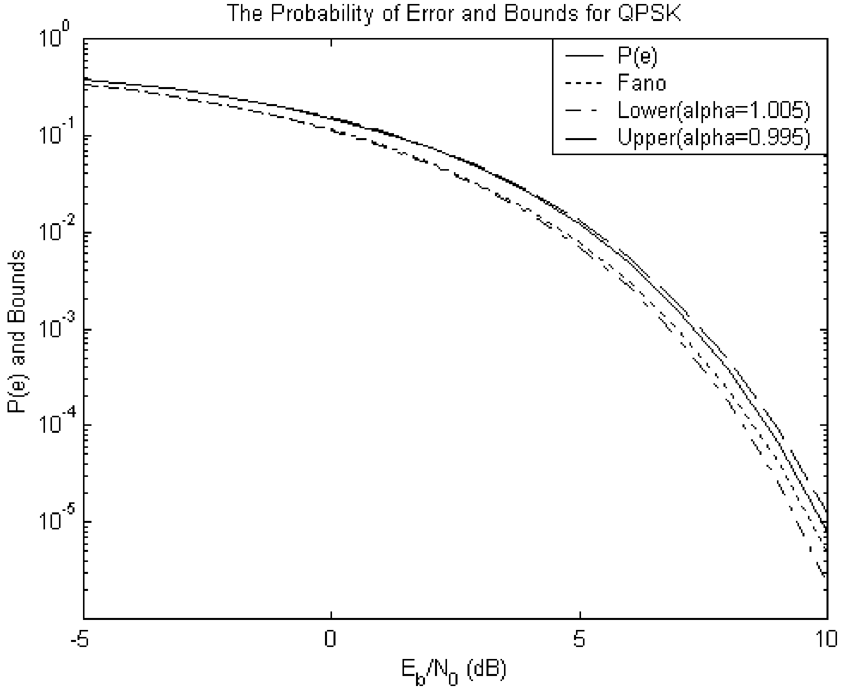
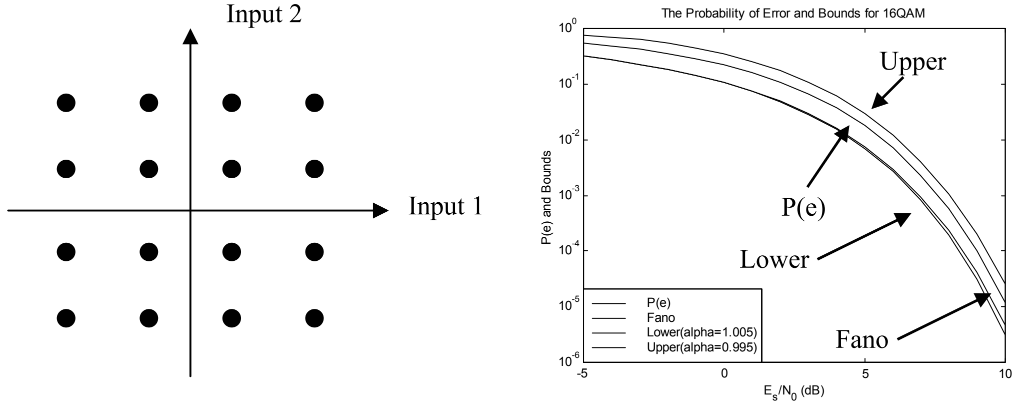


Figure 9-4. Probability of error and its bounds versus bit-energy-to-noise ratio for QPSK

The loss in the denominator of the upper bound increases with increasing number of classes. In order to show this, we extend the previous QPSK modulation scheme to a 16-QAM modulation scheme. Assuming the same AWGN channel model, it is possible to write out analytically the 16x16 confusion matrix in terms of *Q*-functions. For convenience, the 16-QAM constellation is also shown besides the probability of error and its bounds versus the bit-energy-to-noise ratio plot in Figure 9-5. In this example, we observe that, the upper bound becomes looser compared to the QPSK case.



(a) (b)

Figure 9-5. Blind equalization of QAM signals a) 16-QAM constellation; centers of classes in two-dimensional input space b) Probability of error and its bounds versus bit-energy-to-noise ratio for 16-QAM

Finally, we show on the QPSK example, the estimation accuracy for the bounds when a finite number of samples are used to estimate the confusion matrix and the prior probabilities instead of the ideal *Q*-function expressions. By using only a small number of samples, it is possible to get highly accurate estimates of the bounds. In the example below, the average of 1000 Monte Carlo runs, each with 500 randomly selected samples (approximately 125 from each symbol) is presented. As expected, as the bit-energy-to noise-power-ratio increases the estimates become more accurate. Figure 9-6 shows the bias and the standard deviation of the upper bound and the probability of error estimates.

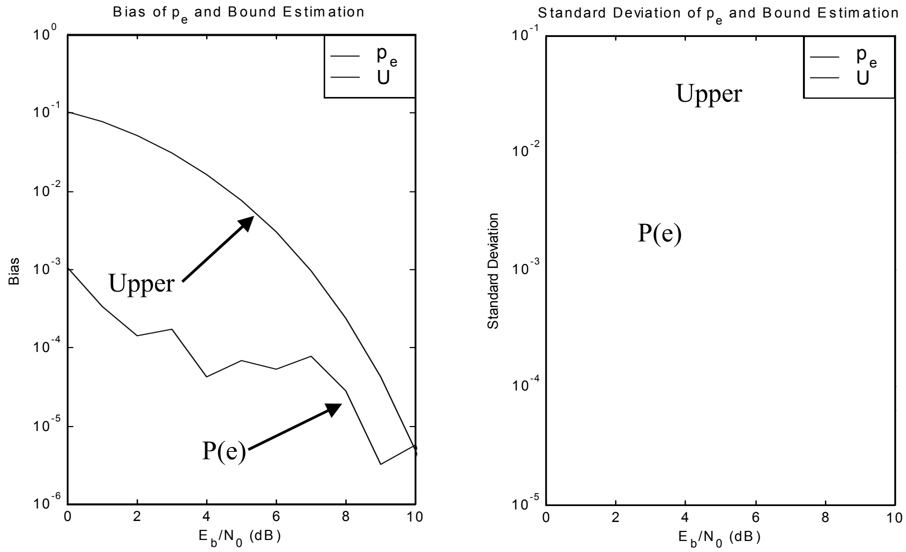


Figure 9-6. Bias and standard deviations in estimating the upper bound and the probability of error for the QPSK example using 500 symbols as a function of bit-energy-to-noise-power ratio

Fano's bound is a widely appreciated inequality that has applications in the proofs of many key theorems in information theory. From a pattern recognition point of view, it is significant in that, it represents the strong connection between classification performance and information. In order to improve our pattern recognition (and in relation to this, feature extraction) capabilities, it is imperative to understand how the information propagation through the classifiers and feature extractors affect the overall performance. Fano's bound provides the attainable limits for performance. However, the bounds we derived in this chapter provide upper bounds for the probability of classification error, which is an important result to evaluate the generalization capabilities. The key conclusion of this chapter is that, by training classifiers to maximize the mutual information between its input and decision classes, it is possible to decrease the probability of error, since both the lower and the upper bound decrease in this situation. However, the entropy of the decisions and the entropy of the *wrong* decisions is important in determining the final performance.

CHAPTER 10
CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

10.1 Summary of Theoretical Results and Their Practical Implications

Adaptive systems are an integral part of signal processing. Adaptive and learning systems also play an important role in many engineering solutions to difficult problems. For decades, second-order statistical criteria had mostly shaped our conception of optimality; this approach flourished mainly because theoretical analysis of second-order performance criteria combined with linear systems is simple. On the other hand, as new engineering problems arose, it became evident that sometimes more than only second-order statistics is either necessary of preferable. Shannon's information theory is probably the most celebrated alternative to second-order statistics in various fields of engineering. Since the aim of information theory is to quantify the *information content* of events and signals, it also provides an intellectually appealing insight and an intuitive understanding to the problems encountered. Shannon's information theoretic quantities like entropy (discrete and differential) and mutual information has already been used by researchers in many contexts of signal processing and even adaptation and learning. However, these attempts were mostly application oriented, and tried to take advantage of the adaptive system topology to simplify or eliminate the direct evaluation of the information theoretic quantities. Some approaches, which tried polynomial expansion estimates of the probability distributions involved in the information theoretic performance measures, suffered from the robustness and data efficiency point-of-view.

The terminology *information theoretic learning* was introduced by Principe *et al.* to signify the optimization of an adaptive system from a finite number of training samples using various information theoretic criteria. Their nonparametric estimator for the quadratic information potential (as they named it) worked pretty well with many different types of data in various different types of problems. However, their work required the support of a mathematical theory motivating and unifying their approaches and observations. This research had started with this motivation and successfully filled in the gaps in the theory behind the experimental observations of Principe *et al.*

A brief overview of the state-of-the-art at the beginning of this research had already been presented in Chapter 1 with appropriate references to previous works of many other researchers, who motivated the use of information theoretic criteria in adaptive system training.

In Chapter 2, the general problem of entropy estimation was addressed; a brief literature survey was portrayed in the form of a unifying perspective that collects various different approaches under appropriate descriptive titles. Following the historical background on entropy estimation, we presented an extended nonparametric estimator for Renyi's entropy (which also covers Shannon's entropy) that reduced to the quadratic entropy estimator for the specific choice of Gaussian kernels, already proposed by Principe *et al.* Chapter 2 continued to elaborate on the mathematical properties of the extended entropy estimator setting forth the conditions for the choice of the kernel functions and founding the necessary theoretical block that would enable the use of itself in the following signal processing and adaptation scenarios. The approach taken in the derivation of the entropy estimator was then extended to the nonparametric estimation of

Renyi's divergence (which includes the Kullback-Leibler divergence as a special case), therefore the estimation of Renyi's mutual information. The quadratic information potential estimator had very interesting properties and a strong analogy to interacting particle in physics, as have been noticed earlier by Principe *et al.*

In Chapter 3, we elaborated on this connection between the generalized information potential (corresponding to different orders of Renyi's entropy) extending the definition of information force to other orders and establishing their relationship with their quadratic counterparts. This investigation revealed invaluable insights on the mechanism of *information theoretic learning* and further inspired us to redefine the principle of minimum or maximum energy learning starting from the basic principles of potential fields, forces and interacting particles. In this point-of-view, we introduced the idea of regarding the training samples as particles that emanate a pre-defined potential field, thus affect the dynamics of the others. We showed that the information potential and the information forces, and the traditional mean-square and other higher-order moments-based criteria, could all be expressed as special cases of this principle, corresponding to different choices of the potential function. Encouraged by the strong link, a backpropagation-of-forces algorithm that would enable efficient implementation of this learning principle in multiplayer perceptrons was introduced as a generalization to the well-known backpropagation-of-errors algorithm.

In Chapter 4, we showed the strengths of the proposed entropy estimator and the use of entropy in general for supervised adaptation of learning systems. This chapter introduced the concept of *minimum error entropy learning* to the field of supervised training, an area that is dominated by the mean-square-error criterion. We established the applicability of the error-entropy criterion in these problems and showed its usefulness in a variety of data sets and problems ranging from chaotic time-series prediction to nonlinear system identification. An interesting relationship between the nonparametric entropy cost function and convolution smoothing of global optimization had emerged from the formulation and was tested in simulations. Although a full proof that shows there is a one-to-one correspondence between the two approaches has not yet been discovered (nor a proof that disproves), results (obtained by myself and other students applying the same cost function to other problems) encourage the further investigation of this issue. Besides many successful applications of the minimum error entropy criterion to supervised learning, Chapter 4 included an effective initialization algorithm for multilayer perceptrons (for use with this criterion) and an analysis of the eigenstructure of the error entropy criterion in the vicinity of the optimal solution.

Chapter 5 was devoted to the application of Renyi's entropy and the associated nonparametric estimator to the problem of independent component analysis. We proposed an effective criterion to solve this problem, which was tested in instantaneous, linear blind source separation scenarios against benchmark algorithms including FastICA and InfoMax. Monte Carlo simulations pointed out that the proposed algorithm was more robust and data efficient than the competing algorithms. An analysis of the performance of the algorithm for various entropy order selections showed that for super-Gaussian signals entropy orders greater than two, for sub-Gaussian signals entropy orders smaller than two could be preferred to improve performance slightly.

Application of the proposed entropy estimator to blind deconvolution and blind equalization was treated in Chapter 6. After a brief discussion and description of the blind deconvolution problem, we showed how the proposed estimator could be used in solving this problem (along with theoretical motivations for Renyi's entropy) and presented a series of successful simulations for blind deconvolution and equalization. The blind equalization example concentrated on a digital communications QPSK-modulation scheme and used an alternative cost function similar to that of the constant-modulus principle, but inspired by the minimum error entropy approach.

All applications considered this far used the batch-mode training approach and due to the $O(N^2)$ complexity of the estimator with respect to the number of samples, this exhibited a significant problem in terms of the computational time required to complete the training tasks. In order to tackle this problem, we followed Widrow's lead as in his derivation of the LMS algorithm, and motivated by the stochastic gradient idea, we derived two alternative stochastic gradient expressions for Renyi's (and Shannon's) entropy in Chapter 7. We established an interesting link between Hebbian learning and the *stochastic information gradient*, as we named it, calling attention to the possibility of an information theoretic learning process persisting in biological adaptive systems. We showed the entropy maximization capability of the stochastic gradient in a simple constructed problem and showed successful applications to supervised learning (using the minimum error entropy principle), blind source separation, and blind deconvolution. Comparisons with other on-line blind source separation algorithms revealed the superiority of the stochastic information gradient. Finally in this chapter, we showed that this stochastic gradient is not only associated with entropy, but is also related to the derivative of the signal under consideration (when the argument of the entropy is deterministic). This motivated us to propose an extension to the traditional mean-square-error criterion for on-line supervised adaptation problems. This approach incorporates the first and possibly higher order derivatives of the error signal into the cost function trying to account for time-varying environments. We also provided an RLS-type algorithm to train linear adaptive systems under the extended criterion, which includes the first order derivative of the error.

Chapter 8 was mainly motivated by the success of the stochastic information gradient and aimed to improve its performance by smoothing its learning curve. In order to achieve this goal, we derived two recursive nonparametric quadratic entropy estimators, one an exact recursion that provides the exact estimate given by the batch estimator, and one a forgetting recursion that incorporates the advantages of a forgetting factor for successful entropy tracking in nonstationary environments. The gradient of the latter is named the recursive information gradient and the stochastic information gradient is shown to be a special case of this corresponding to zero memory (as expected). In this chapter, we also investigated the affect of the three design parameters, namely the forgetting factor, the kernel size and the window length, on the convergence properties of the recursive entropy estimator through simulations. These simulations were also successful demonstrations of the tracking capabilities and the accuracy of the recursive estimators.

Finally in Chapter 9, although not directly related to the previous chapters of the dissertation, we presented an extension to Fano's bound, a well-known result in information theory that links classification performance to the amount of information transferred through the classifier. This extension, based on Renyi's definitions of entropy and mutual information, provided an upper bound and a lower bound (of which the

Fano's bound was the tightest) for the misclassification probability of a given classifier. These bounds proved the common intuition (accepted rather heuristically) in the pattern recognition community, which states that any feature extraction or classification process must transmit as much information as possible from its input space to its output space. We also showed the application of the bounds to a number of classifiers and comparisons to other similar bounds derived using Renyi's entropy.

10.2 Future Research Directions

All these aspects of the current research contribute to the general mathematical theory of *information theoretic learning*, a terminology that we use to describe the nonparametric optimization of adaptive system parameters through the use of information theoretic performance criteria. We recognize that the theory of learning is mostly concentrated around asymptotic results, which are valid if the number of samples approach infinity. On the other hand, a theory of learning from finite number of samples (as in structural risk minimization in pattern recognition) is yet to evolve. Although asymptotic results are necessary to identify the usefulness of the proposed learning approaches, algorithms, topologies, and criteria that extract the most possible information relevant to the desired solution from a given set of finite training samples is really what's necessary for our purposes. This last statement summarizes our long-term intentions. As a short-term research objective to advance the state-of-the-art of information theoretic learning, I would suggest the following:

- Explore the relationship between the proposed minimum error entropy criterion and convolution smoothing. This might provide a valuable general-purpose global information theoretic optimization algorithm to train (supervised or unsupervised) nonlinear adaptive systems for a variety of applications. Global optimization of adaptive system weights is an important issue.

- Investigate various choices of the potential function in the minimum/maximum energy training principle (corresponds to different choices of the kernel function in the information theoretic learning special case). Try determining optimal choices for various applications and data types.

- Investigate noise and outlier rejection capabilities of entropy and other information theoretic performance indices in the finite sample situation. This is imperative for successful generalization of trained networks.

- Investigate extensions/fusion of the criteria and topologies presented for blind source separation and blind deconvolution problems to solve the generalized convolutional mixture case in blind source separation. Extension to instantaneous nonlinear blind source separation is trivial as long as we know the desired source pdfs. The idea is similar to the maximum entropy blind deconvolution approach.

- Investigate the possibility of obtaining recursive estimates for other entropy orders. Compare performances and determine gains of using these alternative entropy orders. (These should be similar to our conclusions on the relationships between different orders of information forces.)

- Investigate the extended criteria involving the derivatives of the error signal under noisy situations. Determine in detail the advantages and disadvantages of this configuration for adapting filters on-line.

- Investigate the possibility of using the extended bounds as a means of providing confidence intervals for classification error probability.

APPENDIX A
SHANNON'S ERROR ENTROPY AND KULLBACK-LEIBLER DIVERGENCE

The derivation that is presented in this appendix is due to Dr. Craig Fancourt. He had used this as part of a proof that shows minimum error entropy is related to the K-L divergence, therefore is a maximum likelihood approach. Suppose that the output of a linear or nonlinear nonlinear adaptive system is expressed as the sum of a deterministic part and a probabilistic error

$$d = f(x) + e \tag{A.1}$$

where *x* is the input to the adaptive system *f*(.), and *d* is the desired output. Let $p_{xd}(x,d)$ be the actual joint density of the input and the desired signals. Let $\tilde{p}_{xd,w}(x,d)$ be the approximation to the actual joint density (given by the joint density of the input and the system output) for a given set of weights *w*. Then, notice that we have the following identities:

$$\tilde{p}_{xd,w}(x,d) = \tilde{p}_{xd,w}(d \mid x)p(x) \tag{A.2}$$

$$\tilde{p}_{xd,w}(d \mid x) = p_{e,w}(d - f(x)) \tag{A.3}$$

where $p_{e,w}(.)$ is the error distribution for a given weight matrix. Minimizing Shannon's error entropy is

$$\min_{w} H_S(e) = -\int p_{e,w}(e)\log p_{e,w}(e)de \tag{A.4}$$

Substituting Eq. (A.3) in Eq. (A.4), we see that this is equivalent to (switching to expectation from integrals)

$$\begin{aligned}\min_{w} J &= -E[\log \tilde{p}_{xd,w}(d \mid x)] - E[\log p(x)] \\ &= -\iint p_{xd}(x,d) \log \tilde{p}_{xd,w}(x,d) dxdd\end{aligned} \tag{A.5}$$

We can add terms that are constant in terms of the system weights.

$$\begin{aligned}\min_{w} J &= -\iint p_{xd}(x,d) \log \tilde{p}_{xd,w}(x,d) dxdd \\ &\quad + \iint p_{xd}(x,d) \log p_{xd}(x,d) dxdd \\ &= -\iint p_{xd}(x,d) \log \frac{\tilde{p}_{xd,w}(x,d)}{p_{xd}(x,d)} dxdd\end{aligned} \tag{A.6}$$

We recognize this last as the K-L divergence between the actual and the approximated joint pdfs. Thus, we conclude that minimizing Shannon's definition of the entropy of error is equivalent to minimizing the divergence between the joint pdfs of input-desired and input-output signals.

APPENDIX B
PROOF FOR NOISE REJECTION OF ERROR ENTROPY FOR ADALINE

Assume that a clean desired signal is generated by $\bar{d} = g(x\ ; w^*)$, where $g(.;w)$ is the adaptive function approximator, and the noisy desired signal is obtained from this signal with $d = \bar{d} + n$. Let the adaptive system output be $y = g(x\ ; w)$ for an arbitrary set of weights in w. Also let $p_n(\eta)$ be the zero-mean noise pdf for n and $p_x(\xi)$ be the pdf of the input signal x. Suppose that the conditional pdf of $\bar{d}$ given x is $p_{\bar{d}|x}(\bar{\delta} \mid \xi\ ; w^*)$ and the noise is independent from the input. Then the conditional pdf of noisy desired given the input is

$$p_{d|x}(\delta \mid \xi\ ; w^*) = p_{\bar{d}|x}(\delta \mid \xi\ ; w^*) * p_n(\delta) \tag{B.1}$$

The error is defined as

$$e(x\ ; w, w^*) = d - y = [g(x\ ; w^*) - g(x\ ; w)] + n \overset{\Delta}{=} m(x\ ; w, w^*) + n \tag{B.2}$$

For an ADALINE structure, the mapping g is given by $g(x\ ; w) = w^T x$, therefore $m(x\ ; w, w^*) = (w^* - w)^T x$. The pdf of m becomes

$$p_{m|x}(\mu \mid \xi\ ; w, w^*) = p_{\bar{d}|x}(\mu \mid \xi\ ; w^* - w) \tag{B.3}$$

Using this, we write the conditional pdf of the error as

$$p_{e|x}(\varepsilon \mid \xi\ ; w, w^*) = p_{m|x}(\varepsilon \mid \xi\ ; w, w^*) * p_n(\varepsilon) \tag{B.4}$$

The probability of error is written as the product of this conditional pdf and the input pdf.

$$\begin{aligned} p_e(\varepsilon) &= p_{e|x}(\varepsilon \mid \xi \ ;w,w^*)p_x(\xi) \\ &= \left[p_{m|x}(\varepsilon \mid \xi \ ;w,w^*) * p_n(\varepsilon)\right]p_x(\xi) \\ &= \left[p_{m|x}(\varepsilon \mid \xi \ ;w,w^*)p_x(\xi)\right] * p_n(\varepsilon) \\ &= \left[p_{\bar{d}|x}(\varepsilon \mid \xi \ ;w^* - w)p_x(\xi)\right] * p_n(\varepsilon) \end{aligned} \tag{B.5}$$

Notice that if w=w^* the error distribution becomes $p_e(\varepsilon) = \delta(\varepsilon) * p_n(\varepsilon) = p_n(\varepsilon)$, thus the error entropy is equal to the noise entropy. Otherwise, the error probability is a convolution of the noise pdf with some other pdf that depends on the current weight vector and the optimal weight vector. In that case, we know that the entropy of the error will be greater than the entropy of noise. Are there any other weight vectors that may lead to a δ-distributed error? In the ADALINE case, as long as the number of training samples is greater than or equal to the number of weights, the answer is 'no', because the weight vector that yields zero error over all samples is determined as the solution to a linear system of equations, and these have unique solutions. This proves that the actual weight vector w^* is the only global minimum of the minimum error entropy criterion, even in the case of noisy desired signal.

APPENDIX C
SIMULTANEOUS PRINCIPAL COMPONENT EXTRACTION

In this appendix, we will present a brief overview of the SIPEX-G algorithm. SIPEX-G is a fast, robust, and accurate PCA algorithm that uses Givens rotations to parameterize the PCA weight matrix, for which we know the optimal solution is orthonormal. This way, the algorithm guarantees that, at every step, the eigenvector estimates are automatically orthonormal, which prevents loosing valuable time and information trying to achieve this property in the weight matrix. The cost function being maximized or minimized is expressed as a combination of the input covariance matrix and the current estimate of the PCA weight matrix to provide accuracy and speed of convergence. Below, we briefly describe the algorithm.

1. Initialize Givens angles, $\Theta = [\theta_{pq}]$.
2. Use the first N>n samples of the input data to obtain an unbiased estimate to the covariance matrix Σ_x .

$$R_x = \frac{1}{N-n}\sum_{k=1}^{N} x_k x_k^T \qquad \text{(C.1)}$$

3. If the input is WSS, use Eq. (C.2), else use a Eq. (C.3) with a fixed forgetting factor.

$$R_x(k) = \frac{k-n-1}{k-n} R_x(k-1) + \frac{1}{k-n} x_k x_k^T \qquad \text{(C.2)}$$

$$R_x(k) = (1-\lambda\ )R_x(k-1) + \lambda\ x_k x_k^T \qquad \text{(C.3)}$$

4. Evaluate the gradient of the cost function with respect to the Givens angles from

$$\frac{\partial J}{\partial \theta_{kl}} = \sum_{o=1}^{n-1} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(R_{oi} \frac{\partial R_{oj}}{\partial \theta_{kl}} + \frac{\partial R_{oi}}{\partial \theta_{kl}} R_{oj} \right) R_{x,ij} \tag{C.4}$$

5. Update the Givens angles using gradient ascent.

$$\Theta(k+1) = \Theta(k) + \eta \ \frac{\partial J}{\partial \Theta} \tag{C.5}$$

6. Go back to step 3 and continue until convergence or as long as new samples continue to arrive.

Detailed discussions on the validity of the cost function and the topology and discussions on the computational complexity of the algorithm are omitted here. Monte Carlo simulation results that show the superiority of SIPEX-G over its gradient-based competitors like Sanger's rule, APEX, and LMSER are also provided [Erd02g].

APPENDIX D
INITIALIZATION ALGORITHM FOR MLPS BASED ON LEAST SQUARES

In batch mode supervised training of nonlinear systems, due to the $O(N^2)$ complexity of the entropy estimator, gradient calculations may require excessive computation time. In order to avoid lengthy training periods, it is necessary to devise a methodology to increase learning speed or to initialize the weights of the adaptive system as near as possible to their optimal values. Especially for the training of MLPs, these two approaches have been extensively exploited by researchers. The studies on the first approach led to the following well-known solutions to the problem of prolonged learning time: the momentum term [Vog88], adaptive step sizes [Jac88], Amari's natural gradient [Ama98], conjugate gradient, which we used in the simulations of the previous section [Mol93, Wat87], exact evaluation of the Hessian [Bis92, Bun93], pseudo-Newton methods [Bar92, Bat92, Fan01, Hag94, Web88, Wat87], random perturbations to the weight updates (inspired by stochastic annealing) [Sty90], genetic algorithms [Ben94], stochastic annealing [Por95]. On the *good* initialization of the weight vector, there has also been some research performed. These include Nguyen and Widrow's approach that assigns each hidden neuron a portion of the range of the output [Ngu90], Drago and Ridella's statistically controlled initialization that aims to prevent saturation of the PEs [Dra92], and heuristic least-squares initialization approaches [Bie93, Yam97], and Castillo *et al.*'s least-square initialization algorithm for a single-layered nonlinear network whose output nonlinearities are also optimized during the training process

[Cas01]. The least squares initialization algorithm that we will provide here can be used to accurately initialize a general *L*-layer MLP with mathematical precision up to the first order of the Taylor series expansion of the nonlinear portions of the network. We will name this approach the *backpropagation of the desired response* as it will be clear in the algorithm that the desired output for the last layer of the MLP is translated back to the preceding layers in a mathematically rigorous fashion. In the following subsections, we will present the derivation for the principled least squares initialization of MLPs, and numerous simulations to verify the performance of this initialization algorithm including chaotic time-series prediction, nonlinear system identification, and classification type training data sets. Although, for analytical reasons, we will use the MSE criterion and try to initialize the weight vector to approximately minimize this criterion, the solution generated by the algorithm could also be used to initialize MLPs that are to be trained according to the MEE principle as experience showed that the optimal solutions of the two criteria are geometrically close to each other in many cases.

D.1 Backpropagating the Desired Signal Through the Layers

Considering the MLP architecture shown in Figure D-1, we notice that there are two parts to backpropagating the desired signal through the layers. Each layer consists of a linear weight matrix and an additive bias term followed by a pre-selected nonlinear mapping, usually chosen to be a sigmoid-type function. For our purposes, the existence of the inverse of this function at every value of its range is necessary. Sigmoid functions, being monotonously increasing, satisfy this requirement. In the following, we designate the output of the l^{th} layer of the MLP by z^l before the nonlinearity and y^l after the nonlinearity. The weight matrix and the bias vector of each layer are designated by W^l

and b^l, respectively. The input to the MLP is called x. Also, n_l is the number of neurons in the corresponding layer and n_0 is the number of inputs. In addition, let N be the number of training samples given in the form $\left(x_t, d_t^L\right)$, where L is the number of layers in the MLP. Finally, we denote by d^l the desired response for the output of the l^{th} layer after the nonlinearity and by $\bar{d}^l$ the desired response for the output before the nonlinearity.

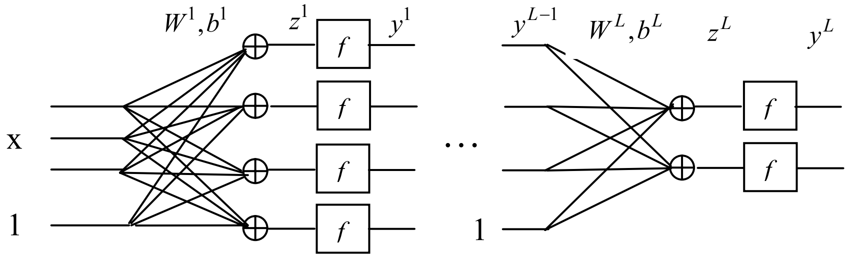


Figure D-1. The MLP structure and variable notations

We will derive the backpropagation of the desired signal algorithm in two parts in accordance with the network structure. These will be the backpropagation through a linear set of weights and the backpropagation through a nonlinearity.

First, we investigate backpropagation through the nonlinearities. Consider a single-layer nonlinear network for which the equations $z=Wx+b$ and $y=f(z)$ define the forward flow of signals, where W,b,f denote the weight matrix, the bias vector, and the output nonlinearity, respectively. Suppose the weighted MSE for a vector output y is the chosen optimization criterion and d is the desired response. Let H be the weighting matrix in the criterion. Then Lemma D.1 describes the backpropagation of the desired response through the output nonlinearity.

Lemma D.1. Let $d, y, z, \bar{d} \in \Re^n$ be the desired and actual outputs, $W \in \Re^{nxm}, b \in \Re^{nx1}$ be the weights, and $f, f^{-1}, f' : \Re^n \rightarrow \Re^n$ be the nonlinearity, its inverse and its derivative. Minimization of the weighted MSE between d and y at the output of the nonlinearity is equivalent (up to first order) to minimizing a weighted MSE between z and $\bar{d} = f^{-1}(d)$, where the inverse function is evaluated at each entry separately. In the latter, each error sample is also weighted according to the value of the derivative of the nonlinearity at the corresponding operating point. Mathematically, this is given by

$$\min_{W,b} E[(d-y)^T H(d-y)] \approx \min_{W,b} E[(f'(\bar{d}).*\bar{\varepsilon})^T H(f'(\bar{d}).*\bar{\varepsilon})] \tag{D.1}$$

where '.* ' denotes the element-wise Hadamard product of the vectors $f'(\bar{d})$ and $\bar{\varepsilon} = \bar{d} - z$.

Proof. Using the fact that $y=f(z)$ and $d = f(\bar{d})$, we can write

$$\min_{W,b} E\left[(d-y)^T H(d-y)\right] = \min_{W,b} E\left[(f(\bar{d}) - f(z))^T H(f(\bar{d}) - f(z))\right] \tag{D.2}$$

Let $\bar{d} = f^{-1}(d)$ be the desired response we seek for z and $\bar{\varepsilon} = \bar{d} - z$, be the error before the nonlinearity. If $\text{var}(\|\bar{\varepsilon}\|)$ is small, then we can use the following first order Taylor series approximation on each component of the output vector.

$$f(z) = f(\bar{d} - \bar{\varepsilon}) \approx f(\bar{d}) - f'(\bar{d}).*\bar{\varepsilon} \tag{D.3}$$

Substituting this in Eq. (D.2), we obtain

$$\min_{W,b} E\left[(d-y)^T H(d-y)\right] \approx \min_{W,b} E\left[(f'(\bar{d}).*\bar{\varepsilon})^T H(f'(\bar{d}).*\bar{\varepsilon})\right] \tag{D.4}$$

which is the result we seek.

Thus, we conclude that, to backpropagate the desired signal through a nonlinearity, we evaluate the inverse of the nonlinear function at the value of the desired signal after the nonlinearity. The weights then must be optimized according to the criterion given in Eq. (D.1). The scaling term $f'(\bar{d})$ ensures that each error sample corresponding to an input-output pair of the training set is magnified appropriately according to the operating point of the nonlinearity at the corresponding value of the desired signal. This result also agrees with our commonsense. For example, when the processing element is near its saturation level, the variance caused by an error sample before the nonlinearity corresponds to a small variance after the nonlinearity. On the other hand, the variance of the error corresponding to a desired signal operating at a large-slope region of the nonlinearity will be magnified by that slope while passing through this nonlinearity. Note that if $\mathrm{var}(\|\bar{d}\|)$ is also small, then since the operating point of the nonlinearity will almost be fixed for all samples, this scaling term becomes unnecessary. All the previous applications of least squares to initialize the weights in MLPs failed to take the variance of d into account, because they simply reflected the desired response to the input of the nonlinearity. This is a poor approximation and as Eq. (D.1) shows, the scaling effect of the nonlinearity on the variance of the error before the nonlinearity should be considered in minimizing the mean-square-error (MSE) at the output of the nonlinearity. In general, for more accurate results one may want to use more terms in the Taylor series expansion, however, this brings in the higher order moments of the error, which prevents us from using the linear least squares fitting algorithms for training.

Secondly, we investigate the backpropagation of the desired output through a linear layer of weights. For this, consider a linear layer whose output is given by $z=Wx+b$ and the desired signal $\bar{d}$ is given for z. In this scheme, we assume the weights W and b are fixed, but the input vector x is the free optimization variable. In the MLP context, x will correspond to the output (after the nonlinearity) of the previous layer. Since the previous layers will produce output vectors only in a bounded subset *D* of the complete vector space due to the limited span of the basis functions, backpropagation of the desired signal of *z* to a desired signal for *x* requires solving a linear weighted least squares problem. The result is summarized in Lemma D.2.

Lemma D.2. Let $d, x \in \Re^{m}, \bar{d}, z \in \Re^{n}$ be the desired signals and corresponding output signals, $W \in \Re^{nxm}, b \in \Re^{nx1}$ be the fixed weight matrix and the bias vector. Minimization of the weighted MSE between $\bar{d}$ and *z* at the output of the linear layer is equivalent to minimizing a weighted MSE between *x* and *d*, i.e., finding the constrained linear least squares solution for the optimal input vector. Mathematically, this is given by

$$\min_{x \in D \subset \Re^{mx1}} E[(\bar{d}-z)^T H(\bar{d}-z)] = \min_{x \in D \subset \Re^{mx1}} E[(d-x)^T W^T HW(d-x)] \qquad \text{(D.5)}$$

Proof. The weighted MSE at the output of the linear layer requires solving

$$\begin{aligned} &\min_{x \in D \subset \Re^{nx1}} E\left[(\bar{d}-z)^T H(\bar{d}-z)\right] = E\left[(\bar{d}^T H\bar{d} - 2\bar{d}^T Hz + z^T Hz)\right] \\ &\equiv \min_{x \in D \subset \Re^{nx1}} -2E\left[\bar{d}^T HWx\right] + 2E\left[b^T HWx\right] + E\left[x^T W^T HWx\right] \end{aligned} \qquad \text{(D.6)}$$

where *H* is again the weight matrix of the weighted MSE criterion. Defining *d* as the least squares solution for the given problem and $\varepsilon \in D' \subset \Re^{nx1}$ as the error between the input *x* and *d* (i.e., $\varepsilon = d\text{-}x$), the above optimization problem becomes equivalent to performing an optimization over the error vector *ε*. The equivalence of these optimization problems is shown in Eq. (D.7).

$$\begin{aligned} &\equiv \min_{\varepsilon \in D'} E\left[(d-\varepsilon)^T W^T HW(d-\varepsilon)\right] - 2E\left[(\bar{d}-b)^T HW(d-\varepsilon)\right] \\ &\quad = E\left[\varepsilon^T W^T HW\varepsilon\right] - 2E\left[d^T W^T HW\varepsilon\right] + 2E\left[(\bar{d}-b)^T HW\varepsilon\right] \\ &\quad = E\left[\varepsilon^T W^T HW\varepsilon\right] \equiv \min_{x \in D} E\left[(d-x)^T W^T HW(d-x)\right] \end{aligned} \qquad \text{(D.7)}$$

Substituting the least squares solution for d in the final expression, the last two terms cancel out to zero leaving the first as the only consideration. The least squares solution is given by the minimum norm solution in general [Lan87]. For the case $n>m$, one can use the well-known form $d=(W^T HW)^{-1}W^{TH}(\bar{d}-b)$.

There are two situations that require special attention at this point.

- If $n \geq m$, then $d=(W^T HW)^{-1}W^T H(\bar{d}-b)$ is the unique weighted least squares solution for the over-determined system of linear equations ($Wd+b=\bar{d}$ with MSE weighting matrix H).
- If $n < m$, then the QR factorization may be used to determine the minimum norm least squares solution for this underdetermined system of linear equations ($Wd+b=\bar{d}$) [Lan87]. Since in this underdetermined case there are infinitely many solutions that yield zero error, the MSE weight matrix H becomes redundant, and any one of the infinitely many solutions can be used.

In both cases, this result tells us that, given a desired signal $\bar{d}^l$ for the linear output z^l of the l^{th} layer, we can translate this signal as a desired signal d^{l-1} for the output (after the nonlinearity) of the previous layer. This value then can be backpropagated through the nonlinearity as described in Lemma D.1.

Once the desired response is backpropagated and it is time to determine the optimal weights or the layer under consideration, it is necessary to solve the following problem.

Problem D.1. Suppose that we are given a linear layer of the form $z=Wx+b$, where $W \in \Re^{nxm}, b \in \Re^{nx1}$. The training samples are given in the input-output configuration as $(x_s, \bar{d}_s)$ $s = 1,...,N$, and also a matrix G=[γ_{ij}] for the weighted least squares is provided. We define the error for every sample of the training data and for every entry of the output vector as

$$\bar{\varepsilon}_{js} = \bar{d}_{js} - z_{js} \quad j = 1,...,n \ , \ s = 1,...,N \tag{D.8}$$

where each output entry is evaluated using

$$z_{js} = b_j + \sum_{i=1}^{N} w_{ji} x_{is} \ , \ j = 1,...,n \ , \ s = 1,...,N \tag{D.9}$$

with w_{ji} denoting the jith entry of the weight matrix, b_j denoting the bias at the jth output node, and x_{is} denoting the ith entry of the input sample x_s. The optimal weights for this layer according to the arguments presented in Lemma D.1 and Lemma D.2 are the solutions to the following minimization problem.

$$\min_{W,b} J = \frac{1}{N} \sum_{s=1}^{N} \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{ij} f'(\bar{d}_{is}) f'(\bar{d}_{js}) \bar{\varepsilon}_{is} \bar{\varepsilon}_{js} \tag{D.10}$$

<u>Solution.</u> In order to solve for the optimal weights for *Problem D.1*, we take the derivatives of the cost function with respect to the weights of the system.

$$\begin{gathered}
\frac{\partial J}{\partial w_{kl}} = \frac{1}{N} \sum_{s=1}^{N} \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{ij} f'(\bar{d}_{is}) f'(\bar{d}_{js}) \left[\bar{\varepsilon}_{is} \frac{\partial \bar{\varepsilon}_{js}}{\partial w_{kl}} + \frac{\partial \bar{\varepsilon}_{is}}{\partial w_{kl}} \bar{\varepsilon}_{js} \right] \\
k = 1,...,n \ , \ l = 1,...,m \\
\frac{\partial J}{\partial b_k} = \frac{1}{N} \sum_{s=1}^{N} \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{ij} f'(\bar{d}_{is}) f'(\bar{d}_{js}) \left[\bar{\varepsilon}_{is} \frac{\partial \bar{\varepsilon}_{js}}{\partial b_k} + \frac{\partial \bar{\varepsilon}_{is}}{\partial b_k} \bar{\varepsilon}_{js} \right] \\
k = 1,...,n
\end{gathered} \tag{D.11}$$

where denoting the Kronecker-delta function with δ_{kj},

$$\frac{\partial \bar{\varepsilon}_{js}}{\partial w_{kl}} = -x_{ls} \delta_{kj} \ , \quad \frac{\partial \bar{\varepsilon}_{js}}{\partial b_k} = -\delta_{kj} \tag{D.12}$$

Equating all the derivatives in Eq. (D.11) to zero and rearranging the terms in order to obtain a square system of linear equations with $n \cdot m + n$ unknown variables yields (D.13). The solution to *Problem D.1* is the solution of this linear system of equations, which could be obtained using a variety of computationally efficient approaches (e.g. Gaussian elimination with pivoting).

$$\begin{aligned}
&\sum_{i=1}^{n} b_i \gamma_{ik} \left[\sum_{s=1}^{N} f'(\bar{d}_{ks}) f'(\bar{d}_{is}) x_{ls} \right] + \sum_{p=1}^{m} \sum_{i=1}^{n} w_{ip} \gamma_{ik} \left[\sum_{s=1}^{N} f'(\bar{d}_{ks}) f'(\bar{d}_{is}) x_{ls} x_{ps} \right] \\
&\qquad = \sum_{i=1}^{n} \gamma_{ik} \left[\sum_{s=1}^{N} f'(\bar{d}_{ks}) f'(\bar{d}_{is}) x_{ls} d_{is} \right] \;,\; k = 1,\ldots,n \;,\; l = 1,\ldots,m \\
&\sum_{i=1}^{n} b_i \gamma_{ik} \left[\sum_{s=1}^{N} f'(\bar{d}_{ks}) f'(\bar{d}_{is}) \right] + \sum_{p=1}^{m} \sum_{i=1}^{n} w_{ip} \gamma_{ik} \left[\sum_{s=1}^{N} f'(\bar{d}_{ks}) f'(\bar{d}_{is}) x_{ps} \right] \\
&\qquad = \sum_{i=1}^{n} \gamma_{ik} \left[\sum_{s=1}^{N} f'(\bar{d}_{ks}) f'(\bar{d}_{is}) d_{is} \right] \;,\; k = 1,\ldots,n
\end{aligned} \tag{D.13}$$

Solving these equations for the variables w_{ip} and b_i, allows one to construct then the optimal weights of the layer under consideration. Notice that the weight matrix G in this optimization problem allows one to take into account the magnifying effect of the succeeding layers on the error of the specific layer. The terms with the derivative of the nonlinearity, on the other hand, allows one to take into consideration the magnifying effect of the slope of the nonlinearity at the end of each layer of the MLP.

D.2 The Backpropagation of the Desired Response Algorithm

Here, we will describe how to apply the procedures devised in Lemma D.1 and Lemma D.2 to the training of a 2-layer MLP (one hidden layer). Although we restrict the algorithm that is presented below to only two layers, the idea can be generalized easily to MLPs with more layers. However, it is known that an MLP with only a single hidden layer that contains a sufficiently large number of hidden PEs can approximate any continuous-differentiable function [Cyb89, Hor91], thus this topology is sufficient in general.

Consider a 2-layer MLP with n_0 inputs, n_1 hidden PEs, and n_2 output PEs. For the sake of generality, we also assume that the output layer contains nonlinearities. We denote the weight matrix and the bias vector of layer l with W^l and b^l, respectively. The output vectors of this layer before and after the nonlinearity are denoted as usual by z^l and y^l, and $\bar{d}^l$ and d^l denote the desired signals for these outputs, respectively. The algorithm for this MLP is as follows:

Algorithm D.1. Backpropagation of the desired response through the layers: Given $\left(x_s, d_s^2\right)$, $s = 1,\ldots,N$. Initialize the weights in W^1, b^1, W^2, b^2 randomly. Evaluate z_s^1, y_s^1, z_s^2, y_s^2 using x_s and the random weights. Set J_{opt} to the MSE between y_s^2 and d_s^2. Set $W_{opt}^1 = W^1$, $b_{opt}^1 = b^1$, $W_{opt}^{21} = W^2$, $b_{opt}^2 = b^2$.

1. Compute $\bar{d}_s^2 = f^{-1}(d_s^2)$, $\forall s$ as the desired signal for z_s^2.
2. Compute $d_s^1 = \left(W^{2T} W^2\right)^{-1} W^{2T}(\bar{d}_s^2 - b^2)$ as the desired signal for y_s^1 (this is for the over-determined case where n_2>n_1, for the underdetermined case, the minimum norm solution could be used).
3. Compute $\bar{d}_s^1 = f^{-1}(d_s^1)$, $\forall s$ as the desired signal for z_s^1.
4. Optimize W^1 and b^1 using the linear least squares equations in Eq. (D.13), using x_s as input samples and $\bar{d}_s^1$ as desired output samples. Use $G = W^{2T} W^2$ as the weighting matrix for weighted MSE criterion (this weight matrix could optionally be chosen as the identity matrix, since in general the weights of the following layers are not

optimal, the use of these as weighting factors for the preceding layers is superfluous; we suggest the *G=I* choice in most cases except for classification type training data).

5. Evaluate z_s^1, y_s^1 using the new values of first layer weights.
6. Optimize W^2 and b^2 using the linear least squares equations in Eq. (D.13), using y_s^1 as input samples and $\bar{d}_s^2$ as desired output samples.
7. Evaluate z_s^2, y_s^2 using the new values of second layer weights.
8. Evaluate the value of *J*, the MSE between y_s^2 and d_s^2. If $J<J_{opt}$, set $J_{opt}=J$, $W_{opt}^1 = W^1$, $b_{opt}^1 = b^1$, $W_{opt}^{21} = W^2$, $b_{opt}^2 = b^2$.
9. Go back to step 2.

The algorithm presented above prefers backpropagating the desired signal all the way to the first layer and then optimizes the weights of the layers sweeping them from the first to the last. Alternatively, first the last layer weights may be optimized, then the desired signal can be backpropagated through that layer using the optimized values of the weights, and so on. Thus, in this alternative algorithm, the layers are optimized sweeping them from the last to the first. Simulations with the latter yield results similar to those obtained by the presented algorithm; therefore, we did not see any reasons to prefer one approach to the other.

Finally, once this algorithm is iterated a number of times (two to five), the weight values that correspond to the smallest MSE error can be assigned as initial conditions to a standard backpropagation algorithm if MSE is to be further used as the optimization criterion. The same weights could also be used as initialization to an MEE training algorithm for the MLP for further information theoretic training.

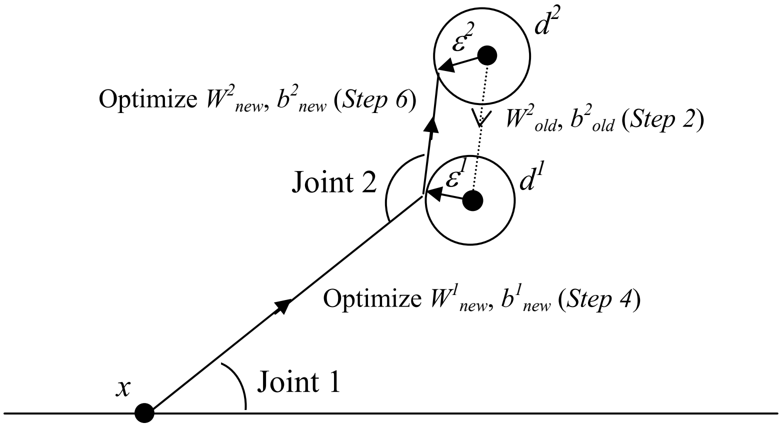

d^2
ε^2
Optimize W^2_{new}, b^2_{new} (*Step 6*)
W^2_{old}, b^2_{old} (*Step 2*)
Joint 2
ε^1
d^1
Optimize W^1_{new}, b^1_{new} (*Step 4*)
x
Joint 1

Figure D-2. Robot arm analogy for the operation of the algorithm

In order to understand better the operation and the behavior of the algorithm (for the two-layer MLP case), consider an analogy to a robotic arm with two joints (depicted in Figure D-2); this analogy is a mechanical equivalent of adapting the bases and the projections at the same time in a two-layer MLP. In this analogy, the weights of the two layers become the angles of the two joints of the robot arm, whose one end is fixed at a given point (a given sample of the input vector) and the other end is trying to reach a desired position in space (the corresponding desired response for the network). The fixed arm lengths signify the limited choice of basis functions and the bounded span of these bases. At every iteration, the robot arm first evaluates the desired position for Joint 1, depending on the current value of Joint 2 (backpropagates the desired response through layers). Then it moves Joint 1 (first layer weights) to bring the first arm as close as possible to its desired location (backpropagated desired response for the first layer). Finally, it moves Joint 2 (second layer weights) to bring the second arm as close as possible to the desired position (actual desired network response). In the following subsection, we investigate the performance of this algorithm in different types of problems involving various data sets. These problems all involve continuously valued desired responses. When applying this algorithm to the training of classification MLPs, it may be necessary to add some small perturbation to the class labels in the training data [Wan99]. A thorough Monte Carlo analysis of the performance of the *backpropagation of the desired response algorithm* is performed using various data types and problems including the generalized XOR classification problem, short-term chaotic time-series prediction problem (Mackey-Glass time-series, laser time-series [Wei84], and Dow Jones closing index time-series), and identification of the realistic nonlinear engine manifold dynamics (based on a car engine [Pow98]), but we omit these results here. These simulations clearly show that the proposed initialization algorithm achieves its goal fast, successfully and efficiently.

APPENDIX E
ERROR DYNAMICS CONTROL APPROACH TO ON-LINE ADAPTATION

In on-line adaptation, it is also possible to adopt the idea of error dynamics control from the *inverse dynamics control* literature, which provides a powerful tool in control theory. The main idea behind *error dynamics control* is to determine the weight updates at every time instant such that the resulting error signal obeys a pre-specified difference equation (for the continuous-time case, this would be a differential equation. In general, one could choose any type of difference equation, as long as it represents a stable system; however, in inverse dynamics control, it is customary to select linear systems of first or second-order. In the second-order linear difference equation case, the desired error dynamics would be

$$e_k + \lambda_1 e_{k-1} + \lambda_2 e_{k-2} = 0 \tag{E.1}$$

where the coefficients λ_i are set such that the difference equation poles form a complex conjugate pair (or two real poles) inside the unit circle for stability. In the simpler first order dynamics case, the one which we will consider in this appendix, the error evolution is simply given by

$$e_k - \lambda\ e_{k-1} = 0 \tag{E.2}$$

where the decay parameter λ is real and between –1 and 1. Consider an ADALINE whose error at time instant k is given by $e_k = d_k - x_k^T w_k$ in terms of its current weight vector. The error dynamics in Eq. (E.2) for this case yield

$$d_k - x_k^T w_k - \lambda\ d_{k-1} + \lambda\ x_{k-1}^T w_{k-1} = 0 \tag{E.3}$$

In order to satisfy this difference equation, w_k must be one of the infinitely many solutions to the equation given in Eq. (E.3), which gives Eq. (E.4) when rearranged.

$$x_k^T w_k = d_k - \lambda\ d_{k-1} + \lambda\ x_{k-1}^T w_{k-1} \tag{E.4}$$

In order to minimize the fluctuations in the weight vector, we would like to have updates that are as small as possible. Therefore, for w_k, from among the infinitely many solution, we select the solution that minimizes the cost function

$$J = (w_k - w_{k-1})^T (w_k - w_{k-1}) \tag{E.5}$$

subject to the constraint Eq. (E.3). Using Lagrange multipliers α, we get the modified cost function as

$$\bar{J} = (w_k - w_{k-1})^T (w_k - w_{k-1}) - \alpha\left(d_k - x_k^T w_k - \lambda\ d_{k-1} + \lambda\ x_{k-1}^T w_{k-1}\right) \tag{E.6}$$

Notice that in Eq. (E.6), besides the Lagrangian, the optimization variable is the vector w_k, and everything else is known constants. Taking the gradient of Eq. (E.6) with respect to the weight vector and the Lagrangians and equating to zero

$$\begin{aligned} \frac{\partial \bar{J}}{\partial w_k} &= 2(w_k - w_{k-1})^T - \alpha\ x_k^T = 0 \\ \frac{\partial \bar{J}}{\partial \alpha} &= d_k - x_k^T w_k - \lambda\ d_{k-1} + \lambda\ x_{k-1}^T w_{k-1} = 0 \end{aligned} \tag{E.6}$$

we can solve for the Lagrangian and the optimal solution of the constrained weight vector. The linear set of equations in Eq. (E.6) yield

$$\begin{aligned} \alpha &= \frac{2}{(x_k^T x_k)}\left(d_k - \lambda\ d_{k-1} + \lambda\ x_{k-1}^T w_{k-1} - x_k^T w_{k-1}\right) \\ w_k &= w_{k-1} + x_k \frac{\left(d_k - \lambda\ d_{k-1} + \lambda\ x_{k-1}^T w_{k-1} - x_k^T w_{k-1}\right)}{(x_k^T x_k)} \end{aligned} \tag{E.7}$$

which provide the weight update equation that minimizes the norm of the update subject to the constraint of desired error dynamics.

Especially interesting is the special case where λ=0. In that case, the weight update equation reduces to

$$w_k = w_{k-1} + x_k \frac{\left(d_k - x_k^T w_{k-1}\right)}{(x_k^T x_k)} \tag{E.8}$$

which resembles the well-known normalized LMS update rule.

APPENDIX F
DERIVATION OF EXTENDED FANO'S BOUNDS USING RENYI'S DEFINITIONS OF JOINT ENTROPY AND MUTUAL INFORMATION

First, consider Renyi's joint entropy. Staring from the definition, and applying Baye's rule and Jensen's inequality, for different values of α, we obtain two inequalities.

$$\begin{aligned}
H_\alpha(W,M) &= \frac{1}{1-\alpha}\log\sum_k\sum_j p^\alpha(w_j,m_k) \\
&= \frac{1}{1-\alpha}\log\sum_k\sum_j p^\alpha(w_j \mid m_k)p^\alpha(m_k) \\
&\underset{\underset{\alpha<1}{\geq}}{\overset{\alpha>1}{\leq}} \sum_k p(m_k)\frac{1}{1-\alpha}\log\sum_j p^\alpha(w_j \mid m_k) \\
&= \sum_k p(m_k)\left[-\log p(m_k)+\frac{1}{1-\alpha}\log\left[\begin{array}{l}\sum_{j\neq k} p^\alpha(w_j \mid m_k) \\ +\,p^\alpha(w_k \mid m_k)\end{array}\right]\right] \\
&= H_S(M)+\sum_k p(m_k)\frac{1}{1-\alpha}\log\left[\begin{array}{l}\sum_{j\neq k} p^\alpha(w_j \mid m_k) \\ +\,p^\alpha(w_k \mid m_k)\end{array}\right] \\
&\underset{\underset{\alpha<1}{\geq}}{\overset{\alpha>1}{\leq}} H_S(M)+\sum_k p(m_k)\left[\begin{array}{l}H_S(e \mid m_k) \\ +\,p(e \mid m_k)\dfrac{1}{1-\alpha}\log\displaystyle\sum_{j\neq k}\left(\frac{p(w_j \mid m_k)}{p(e \mid m_k)}\right)^\alpha\end{array}\right]
\end{aligned} \tag{F.1}$$

Hence, rearranging the terms, we obtain the following inequality

$$\frac{H_\alpha(W,M)-H_S(M)-H_S(e)}{\log(N_c-1)} \leq p_e \leq \frac{H_\beta(W,M)-H_S(M)-H_S(e)}{\min\limits_k H_\beta(W \mid e,m_k)}, \quad \begin{array}{l}\alpha \geq 1 \\ \beta < 1\end{array} \tag{F.2}$$

Now consider Renyi's mutual information. Once again applying Jensen's inequality in two steps, we can obtain the lower and upper bounds for error probability. In the bound for mutual information, Jensen's inequality is applied two times.

$$
\begin{aligned}
I(M;W) &= \frac{1}{\alpha-1}\log\sum_k\sum_j \frac{p^{\alpha}(w_j,m_k)}{p^{\alpha-1}(w_j)p^{\alpha-1}(m_k)} \\
&= \frac{1}{\alpha-1}\log\sum_k\sum_j \frac{p^{\alpha}(w_j\,|\,m_k)p(m_k)}{p^{\alpha-1}(w_j)} \\
&\overset{\alpha>1}{\underset{\alpha<1}{\substack{\le \\ \ge}}} \sum_k p(m_k)\frac{1}{\alpha-1}\log\sum_j p^{\alpha}(w_j\,|\,m_k)p^{1-\alpha}(w_j) \\
&= \sum_k p(m_k)\frac{1}{\alpha-1}\log\begin{bmatrix} \sum\limits_{j\neq k} p^{\alpha}(w_j\,|\,m_k)p^{1-\alpha}(w_j) \\ +p^{\alpha}(w_k\,|\,m_k)p^{1-\alpha}(w_k) \end{bmatrix} \\
&\overset{\alpha>1}{\underset{\alpha<1}{\substack{\le \\ \ge}}} \sum_k p(m_k)\Bigg[p(e\,|\,m_k)\frac{1}{\alpha-1}\log\left(p^{\alpha-1}(e\,|\,m_k)\sum_{j\neq k}\frac{p^{\alpha}(w_j\,|\,m_k)p^{1-\alpha}(w_j)}{p^{\alpha}(e\,|\,m_k)}\right) \\
&\qquad\qquad + (1-p(e\,|\,m_k))\frac{1}{\alpha-1}\log(1-p(e\,|\,m_k))^{\alpha-1}p^{1-\alpha}(w_k)\Bigg]
\end{aligned}
\tag{F.3}
$$

Now, rearranging the terms, and applying Jensen's inequality once more,

$$
\begin{aligned}
I_{\alpha}(M;W) &\overset{\alpha>1}{\underset{\alpha<1}{\substack{\le \\ \ge}}} \sum_k p(m_k)\begin{bmatrix} -H_S(e\,|\,m_k) \\ +p(e\,|\,m_k)\dfrac{1}{1-\alpha}\log\sum\limits_{j\neq k}\dfrac{p^{\alpha}(w_j\,|\,m_k)p^{1-\alpha}(w_j)}{p^{\alpha}(e\,|\,m_k)} \\ -(1-p(e\,|\,m_k))\log p(w_k) \end{bmatrix} \\
&= \sum_k p(m_k)\begin{bmatrix} -H_S(e\,|\,m_k)+p(e\,|\,m_k)\log p(w_k) \\ -\log p(w_k) \\ +p(e\,|\,m_k)\dfrac{1}{1-\alpha}\log\sum\limits_{j\neq k}\dfrac{p^{\alpha}(w_j\,|\,m_k)p^{1-\alpha}(w_j)}{p^{\alpha}(e\,|\,m_k)} \end{bmatrix} \\
&\overset{\alpha>1}{\underset{\alpha<1}{\substack{\le \\ \ge}}} \sum_k p(m_k)\begin{bmatrix} -H_S(e\,|\,m_k)+p(e\,|\,m_k)\log p(w_k)-\log p(w_k) \\ +p(e\,|\,m_k)\sum\limits_{j\neq k}\dfrac{p(w_j\,|\,m_k)}{p(e\,|\,m_k)}\dfrac{1}{1-\alpha}\log\dfrac{p^{\alpha-1}(w_j\,|\,m_k)p^{1-\alpha}(w_j)}{p^{\alpha-1}(e\,|\,m_k)} \end{bmatrix}
\end{aligned}
\tag{F.4}
$$

which, after recollecting terms, becomes equal to

$$
=\sum_k p(m_k)\left[\begin{array}{l} -H_S(e\,|\,m_k)+p(e\,|\,m_k)\log p(w_k)-\log p(w_k) \\ +\,p(e\,|\,m_k)\displaystyle\sum_{j\neq k}\frac{p(w_j\,|\,m_k)}{p(e\,|\,m_k)}\left[\log\frac{p(w_j\,|\,m_k)}{p(e\,|\,m_k)}-\log p(w_j)\right] \end{array}\right]
$$

$$
\begin{aligned}
&=-H_S(e)+\sum_k p(m_k)\left[\begin{array}{l} -\,p(w_k\,|\,m_k)\log p(w_k) \\ +\,p(e\,|\,m_k)\displaystyle\sum_{j\neq k}\frac{p(w_j\,|\,m_k)}{p(e\,|\,m_k)}\log\frac{p(w_j\,|\,m_k)}{p(e\,|\,m_k)} \\ -\displaystyle\sum_{j\neq k}p(w_j\,|\,m_k)\log p(w_j) \end{array}\right] \\
&=-H_S(e)+H_S(W)+\sum_k p(m_k)p(e\,|\,m_k)\sum_{j\neq k}\frac{p(w_j\,|\,m_k)}{p(e\,|\,m_k)}\log\frac{p(w_j\,|\,m_k)}{p(e\,|\,m_k)}
\end{aligned}
\tag{F.5}
$$

Finally, rearranging the terms and substituting appropriate extreme values for the multiplier of p_e, we obtain the following inequality on error probability in terms of Renyi’s mutual information.

$$
\frac{H_S(W)-I_\alpha(W;M)-H_S(e)}{\log(N_c-1)}\leq p_e\leq\frac{H_S(W)-I_\beta(W;M)-H_S(e)}{\min\limits_k H_S(W\,|\,e,m_k)},\quad \begin{array}{l}\alpha\geq 1\\ \beta<1\end{array}
\tag{F.6}
$$

REFERENCES

[Ahm76] I.A. Ahmad, P.E. Lin, "A Nonparametric Estimation of the Entropy for Absolutely Continuous Distributions," IEEE Transactions on Information Theory, vol. 22, pp. 372-375, 1976.

[Ahm89] N.A. Ahmed, D.V. Gokhale, "Entropy Expressions and Their Estimators for Multivariate Distributions," IEEE Transactions on Information Theory, vol. 35, pp. 688-692, 1989.

[Ama85] S. Amari, *Differential–Geometrical Methods in Statistics*, Springer-Verlag, Berlin, 1985.

[Ama96] S. Amari, "Neural Learning in Structured Parameter Spaces–Natural Riemannian Gradient," Proceedings of NIPS'96, pp. 127-133, Denver, CO, 1996.

[Ama98] S. Amari, "Natural Gradient Works Efficiently in Learning," Neural Computation, vol. 10, pp. 251-276, 1998.

[Bar92] E. Barnard, "Optimization for Training Neural Nets," IEEE Transactions on Neural Networks, vol. 2, no. 5, pp. 498-508, 1992.

[Bas78] M. B. Bassat, J. Raviv, "Renyi's Entropy and the Probability of Error," IEEE Transactions on Information Theory, vol. 24, no. 3, pp. 324-330, May 1978.

[Bat92] R. Battiti, "First and Second Order Methods for Learning: Between Steepest Descent and Newton's Method," Neural Computation, vol. 4, no. 2, pp. 141-166, 1992.

[Bec93] C. Beck, F. Schlogl, *Thermodynamics of Chaotic Systems*, Cambridge University Press, Cambridge, 1993.

[Bei01] J. Beirlant, E.J. Dudewicz, L. Gyorfi, E.C. van der Meulen, "Nonparametric Entropy Estimation: An Overview," International Journal of Mathematical and Statistical Sciences, vol. 6, no. 1, pp. 17-39, 1997.

[Bei85] J. Beirlant, M.C.A. Zuijlen, "The Empirical Distribution Function and Strong Laws for Functions of Order Statistics of Uniform Spacings," Journal of Multivariate Analysis, vol. 16, pp. 300-317, 1985.

[Bel95] A. Bell, T. Sejnowski, "An Information-Maximization Approach to Blind Separation and Blind Deconvolution," Neural Computation vol. 7, pp. 1129-1159, 1995.

[Ben94] S. Bengio, Y. Bengio, J. Cloutier, "Use of Genetic Programming for the Search of a New Learning Rule for Neural Networks," Proceedings of the 1st IEEE World Congress on Computational Intelligence and Evolutionary Computation, pp. 324-327, 1994.

[Ben80] A. Benveniste, M. Goursat, G. Ruget, "Robust Identification of a Nonminimum Phase System: Blind Adjustment of a Linear Equalizer in Data Communications," IEEE Transactions on Automatic Control, vol. 25, no. 3, pp. 385-399, 1980.

[Ber00] J.F. Bercher, C. Vignat, "Estimating the Entropy of a Signal with Applications," IEEE Transactions on Signal Processing, vol. 48, no. 6, pp. 1687-1694, 2000.

[Bic83] P.J. Bickel, L. Breiman, "Sums of Functions of Nearest Neighbor Distances, Moment Bounds, Limit Theorems and a Goodness-of-fit Test," Annals of Statistics, vol. 11, pp. 185-214, 1983.

[Bie93] F. Biegler-Konig, F. Barnmann, "A Learning Algorithm for Multilayered Neural Networks Based on Linear Least Squares Problems," Neural Networks, vol. 6, pp. 127-131, 1993.

[Bis92] C.M. Bishop, "Exact Calculation of the Hessian Matrix for the Multilayer Perceptron," Neural Computation, vol. 4, no. 4, pp. 494-501, 1992.

[Bis95] C. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.

[Bos01] R. Boscolo, H. Pan, V.P. Roychowdhury, "Non-Parametric ICA", Proceedings of ICA'01, San Diego, CA, 2001.

[Bun93] W.L. Buntine, A.S. Weigend, "Computing Second Derivatives in Feed-Forward Networks: A Review," IEEE Transactions on Neural Networks, vol. 5, no. 3, pp. 480-488, 1993.

[Cac97] C. Cachin, "Smooth Entropy and Renyi Entropy," in Lecture Notes in Computer Science, vol. 1233, (ed. Walter Fumy), Advances in Cryptology: Eurocrypt'97, Springer-Verlag, 1997, pp. 193-208.

[Cam65] L.L. Campbell, "A Coding Theorem and Renyi's Entropy," Information and Control, vol. 8, pp. 423--429, 1965.

[Car98] J. Cardoso, "Blind Signal Separation: Statistical Principles," Proceedings of IEEE, vol. 86, no. 10, pp. 2009-2025, 1998.

[Cas00] J. Casals, C. Jutten, A. Taleb, "Source Separation Techniques Applied to Linear Prediction", Proceedings of ICA'00, Helsinki, Finland, pp. 193-204, 2000.

[Cas01] E. Castillo, O. Fontenla-Romero, A. Alonso-Betanzos, B. Guijarro-Berdinas, "A Global Optimum Approach for One-Layer Neural Networks," to appear in Neural Computation, 2002.

[Chi01] C.Y. Chi, C.H. Chen, "Cumulant-based Inverse Filter Criteria for MIMO Blind Deconvolution: Properties, Algorithms, and Application to D/CDMA Systems in Multipath," IEEE Transactions on Signal Processing, vol. 49, no. 7, pp. 1282-1299, 2001.

[Cho00] S. Choi, A Cichocki, S. Amari, "Flexible Independent Component Analysis," Journal of VLSI Signal Processing, vol. 26 pp. 25-38, 2000.

[Cla98] R.L. Clark, W.R. Saunders, G.P. Gibbs, *Adaptive Structures: Dynamics and Control*, Wiley, New York, 1998.

[Com94] P. Comon, "Independent Component Analysis, A New Concept?," Signal Processing, vol. 36 pp. 287-314, 1994.

[Cov91] T. Cover, J. Thomas, *Elements of Information Theory*, Wiley, New York, 1991.

[Csi81] I. Csiszár, J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*, Academic Press, New York, 1981.

[Cyb89] G. Cybenko, "Approximation by Superposition of a Sigmoidal Function," Mathematics of Control, Signals, Systems, vol. 2, pp. 303-314, 1989.

[Dec96] G. Deco, D. Obradovic, "An Information-Theoretic Approach to Neural Computing", New York, Springer, 1996.

[Dia01] K.I. Diamantaras, "PCA Neural Models for Blind Signal Separation," Proceedings of IJCNN'01, Washington, DC, 2001.

[Dmi73] Y.G. Dmitrev, F.P. Tarasenko, "On the Estimation Functions of the Probability Density and its Derivatives," Theory of Probability and its Applications, vol. 18, pp. 628-633, 1973.

[Don81] D. Donoho, "On Minimum Entropy Deconvolution," in *Applied Time Series Analysis II*, Academic Press, New York, 1981, pp. 565-609.

[Dra92] G.P. Drago, S. Ridella, “Statistically Controlled Activation Weight Initialization (SCAWI),” IEEE Transactions on Neural Networks, vol. 3, pp. 899-905, 1992.

[Dur98] R. Durbin, S. Eddy, A. Krogh, G. Mitchinson, *Biological Sequence Analysis*, Campbridge University Press, Cambridge, 1998.

[Dur99] R.J. Duro, J.S. Reyes, “Discrete-time Backpropagation for Training Synaptic Delay-based Artificial Neural Networks,” IEEE Transactions on Neural Networks, vol. 10, no. 4, pp. 779-789, 1999.

[Edm96] W. Edmonson, K. Srinivasan, C. Wang, J. Principe. “A Global Least Square Algorithm for Adaptive IIR Filtering,” IEEE Transactions on Circuits and Systems, vol. 45 no.3, pp. 379-384, 1996.

[Erd02a] D. Erdogmus, A.U. Genc, J.C. Principe, “A Neural Network Perspective to Extended Luenberger Observers,” to appear in Institute of Measurement and Control Special Feature on Recent Advances in Neural Networks, Part 2, 2002.

[Erd02b] D. Erdogmus, K.E. Hild II, J.C. Principe, "Blind Source Separation Using Renyi's Marginal Entropy", to appear in Neurocomputing Special Issue on Blind Source Separation and Independent Component Analysis, 2002.

[Erd00a] D. Erdogmus, J.C.Principe, “Comparison of Entropy and Mean Square Error Criteria in Adaptive System Training Using Higher Order Statistics”, Proceedings of ICA’00, pp. 75-80, Helsinki, Finland, 2000.

[Erd01a] D. Erdogmus, J.C. Principe, “Information Transfer Through Classifiers and its Relation to Probability of Error,” Proceedings of IJCNN’01, Washington, DC, 2001.

[Erd01b] D. Erdogmus, J.C. Principe, “Convergence Analysis of the Information Potential Criterion in ADALINE Training,” Proceedings of NNSP’01, pp. 123-132, Falmouth, MA, 2001.

[Erd01c] D. Erdogmus, J.C. Principe, “An On-Line Adaptation Algorithm for Adaptive System Training with Minimum Error Entropy: Stochastic Information Gradient,” Proceedings of ICA’01, pp. 7-12, San Diego, CA, 2001.

[Erd02c] D. Erdogmus, J.C.Principe, “An Entropy Minimization Algorithm for Supervised Training of Nonlinear Systems,” to appear in IEEE Transactions on Signal Processing, 2002.

[Erd02d] D. Erdogmus, J.C. Principe, "Generalized Information Potential Criterion for Adaptive System Training," to appear in IEEE Transactions on Neural Networks, 2002.

[Erd02e] D. Erdogmus, J.C. Principe, "Insights on the Relationship Between Probability of Misclassification and Information Transfer Through Classifiers," to appear in International Journal of Computers, Systems, and Signals, 2002.

[Erd02f] D. Erdogmus, J.C Principe, "Lower and upper Bounds for Misclassification Probability Based on Renyi's Information," accepted to Journal of VLSI Signal Processing special Issue on Wireless Communications and Blind Signal Processing, 2002.

[Erd02g] D. Erdogmus, Y.N. Rao, J.C. Principe, J. Zhao, K.E. Hild II, "Simultaneous Extraction of Principal Components Using Givens Rotations and Output Variances," accepted to ICASSP'02, Orlando, FL, 2002.

[Fan01] C. Fancourt, J.C. Principe, "Optimization in Companion Search Spaces: The Case of Cross-Entropy and the Levenberg-Marquardt Algorithm," Proceedings of ICASSP'01, Salt Lake City, 2001.

[Fan61] R.M. Fano, *Transmission of Information: A Statistical Theory of Communications*, MIT Press, New York, 1961.

[Far98] B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications*, Wiley, New York, 1998.

[Fed94] M. Feder, N. Merhav, "Relations Between Entropy and Error Probability," IEEE Transactions on Information Theory, vol. 40, no. 1, pp. 259-266, 1994.

[Fen97] X. Feng, K. Loparo, Y. Fang, "Optimal State Estimation for Stochastic Systems: An Information Theoretic Approach", IEEE Transactions on Automatic Control, vol. 42, no. 6, pp. 771-785, 1997.

[Fis97] J.W. Fisher, *Nonlinear Extensions to the Minimum Average Correlation Energy Filter*, PhD dissertation, University of Florida, 1997.

[Fis00] J.W. Fisher, A. Ihler, P. Viola, "Learning Informative Statistics: A Nonparameteric Approach", Proceedings of NIPS'00, pp. 900-906, 2000.

[Fu70] K. Fu, "Statistical Pattern Recognition," in *Adaptive, Learning and Pattern Recognition* Systems, (Ed. Mendel and Fu), Academic Press, New York, 1970, pp. 35-76.

[Fuk72] K. Fukunaga, *An Introduction to Statistical Pattern Recognition*, Academic Press, New York, 1972.

[Gac94] P. Gacs, "The Boltzmann Entropy and Randomness Tests," Proceedings of Physics and Computation, pp. 209 – 216, 1994.

[Gal68] R.G. Gallager, *Information Theory and Reliable Communication*, John Wiley & Sons Inc, New York, 1968.

[God80] D.N. Godard, "Self-recovering Equalization and Carrier Tracking in Two-dimensional Data Communication Systems," IEEE Transactions on Communications, vol. 28, no. 11, pp. 1867-1875, 1980.

[Goo84] G.C. Goodwin, K.S. Sin, *Adaptive Filtering, Prediction and Control*, Prentice Hall, Upper Saddle River, New Jersey, 1984.

[Gyo87] L. Gyorfi, E.C. van der Meulen, "Density-Free Convergence Properties of Various Estimators of Entropy," Computational Statistics and Data Analysis, vol. 5, pp. 425-436, 1987.

[Gyo89] L. Gyorfi, E.C. van der Meulen, "An Entropy Estimate Based on a Kernel Density Estimation," in Limit Theorems in Probability and Statistics, (I. Berkes, E. Csaki, P. Revesz eds.), North Holland, 1989, pp. 229-240.

[Gyo90] L. Gyorfi, E.C. van der Meulen, "On Nonparametric Estimation of Entropy Functionals," in Nonparametric Functional Estimation and Related Topics, (G. Roussas ed.), Kluwer Academic Publisher, Amsterdam, 1990, pp. 81-95.

[Hag94] M.T. Hagan, M.B. Menhaj, "Training Feedforward Networks with the Marquardt Algorithm," IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989-993, 1994.

[Hal84] P. Hall, "Limit Theorems for Sums of General Functions of *m*-Spacings," Mathematical Proceedings of the Cambridge Philosophical Society, vol. 96 pp. 517-532, 1984.

[Han94] T.S. Han, S. Verdu, "Generalizing the Fano Inequality," IEEE Transactions on Information Theory, vol. 40, no. 4, pp.1247-1251, 1994.

[Hay84] S. Haykin, *Introduction to Adaptive Filters*, MacMillan, New York, 1984.

[Hay94] S. Haykin (ed.), *Blind Deconvolution*, Prentice-Hall, Upper Saddle River, New Jersey, 1994.

[Hay96] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, New Jersey, 1996.

[Hay99] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed., Prentice Hall, Upper Saddle River, New Jersey, 1999.

[Hay00a] S. Haykin (ed.), *Unsupervised Adaptive Filtering, Volume 1: Blind Source Separation*, John Wiley & Son, Inc., New York, 2000.

[Hay00b] S. Haykin (ed.), *Unsupervised Adaptive Filtering, Volume 2: Blind Deconvolution*, John Wiley & Son, Inc., New York, 2000.

[Hay98] S. Haykin, J.C. Principe, "Dynamic Modeling with Neural Networks", in IEEE Signal Processing Magazine, vol. 15, no. 3, pp. 66-81, 1998.

[Heb49] D.O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*, Wiley, New York, 1949.

[Hil01a] K.E. Hild II, D. Erdogmus, J.C. Principe, "Blind Source Separation Using Renyi's Mutual Information," IEEE Signal Processing Letters, no. 8, pp. 174-176, 2001.

[Hil01b] K.E. Hild II, D. Erdogmus, J.C. Principe, "On-Line Minimum Mutual Information Method for Time-Varying Blind Source Separation," Proceedings of ICA'01, pp. 126-131, San Diego, CA, 2001.

[Hil02] K.E. Hild II, D. Erdogmus, J.C. Principe, "Blind Source Separation of Time-Varying, Instantaneous Mixtures Using an On-Line Algorithm," accepted to ICASSP'01, Orlando, FL, 2002.

[Hor91] K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks," Neural Networks, vol. 4, pp. 251-257, 1991.

[Hyv99a] A. Hyvarinen, "Fast and Robust Fixed-point Algorithms for Independent Component Analysis," IEEE Transactions on Neural Networks, vol. 10, pp. 626-634, 1999.

[Hyv99b] A. Hyvarinen, "Survey on Independent Component Analysis," Neural Computation, Surveys, vol. 2, pp. 94-128, 1999.

[Hyv01] A. Hyvarinen, J. Karhunen, E. Oja, *Independent Component Analysis*, Wiley, New York, 2001.

[Iva81] A.V. Ivanov, A. Rozhkova, "Properties of the Statistical Estimate of the Entropy of a Random Vector with a Probability Density," Problems of Information Transmission, vol. 17, pp. 171-178, 1981.

[Jac88] R.A. Jacobs, "Increased Rates of Convergence Through Learning Rate Adaptation," Neural Networks, vol. 1, no. 4, pp. 295-308, 1988.

[Jan00] Y. Janghoon, C.L. Nikias, "The Blind Deconvolution of the Multi-hannel Based on the Higher Order Statistics," Conference Record of the 34th Asimolar Conference on Signals, Systems, and Computers, vol. 2, pp. 1192-1196, 2000.

[Joe89] H. Joe, "On the Estimation of Entropy and Other Functionals of a Multivariate Density," Annals of the Institute of Statistical Mathematics vol. 41, pp. 683-697, 1989.

[Kap95] D. Kaplan, L. Glass, *Understanding Nonlinear Dynamics*, Springer-Verlag, New York, 1995.

[Kap92] J. Kapur, H. Kesavan, *Entropy Optimization Principles and Applications*, Associated Press, New York, 1992.

[Koz87] L.F. Kozachenko, N.N. Leonenko, "Sample Estimate of Entropy of a Random Vector," Problems of Information Transmission, vol. 23, pp. 95-101, 1987.

[Kul68] S. Kullback, *Information Theory and Statistics*, Dover Publications, Inc., New York, 1968.

[Lan87] S. Lang, *Linear Algebra*, 3rd ed., Springer-Verlag, New York, 1987.

[Lin88] R. Linsker, "Towards an Organizing Principle for a Layered Perceptual Network," Proceedings of NIPS'88, pp 485-494, 1988.

[Lue73] D.G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley Pub. Co., Boston, Massachussetts, 1973.

[Mit75] D.P. Mittal, "On Additive and Non-additive Entropies," Kybernetika, vol. 11, no. 4, pp. 271-276, 1975.

[Mol93] M. Moller, *Efficient Training of Feedforward Neural Networks*, Ph.D. Thesis, Aarhus University, Denmark, 1993.

[Nar89] K.S. Narendra, A.M. Annaswamy, *Stable Adaptive Systems*, Prentice Hall, Upper Saddle River, New Jersey, 1989.

[Nik93] C.L. Nikias, A.P. Petropulu, *Higher-order Spectra Analysis: A Nonlinear Signal Processing Framework*, Prentice Hall, Upper Saddle River, New Jersey, 1993.

[Ngu90] D. Nguyen, B. Widrow, “Improving the Learning Speed of 2-layer Neural Networks by Choosing Initial Values of the Adaptive Weights,” Proceedings of IJCNN’90, vol. 3, pp. 21-26, 1990.

[Oja83] E. Oja, *Subspace Methods of Pattern Recognition*, Wiley, New York, 1983.

[Par67] E. Parzen, “On Estimation of a Probability Density Function and Mode”, in *Time Series Analysis Papers*, Holden-Day, Inc., San Diego, California, 1967.

[Poo95] H.V. Poor, S. Verdu, “A Lower Bound on the Probability of Error in Multihypothesis Testing,” IEEE Transactions on Information Theory, vol. 41, no. 6, pp. 1992-1994, 1995.

[Por95] V.W. Porto, D.B. Fogel, “Alternative Neural Network Training Methods [Active Sonar Processing],” IEEE Expert, vol. 10, no. 3, pp. 16-22, 1995.

[Pow98] J.D. Powell, N.P. Fekete, C-F. Chang, “Observer-Based Air-Fuel Ratio Control,” IEEE Control Systems Magazine, vol. 18, no. 5, pp. 72-83, 1998.

[Pri99] J.C. Principe, N. Euliano, C. Lefebvre, *Neural and Adaptive Systems: Fundamentals Through Simulations*, Wiley, New York, 1999.

[Pri00a] J.C. Principe, J.W. Fisher, D. Xu, “Information Theoretic Learning,” in *Unsupervised Adaptive Filtering*, S. Haykin Editor, John Wiley & Sons, New York, 2000, pp.265-319.

[Pri00b] J.C. Principe, D. Xu, Q. Zhao, J. Fisher, “Learning from Examples with Information Theoretic Criteria,” Journal of VLSI Signal Processing Systems, Special Issue on Neural Networks, pp. 61-77, 2000.

[Qur85] S.U.H. Qureshi, “Adaptive Equalization,” Proceedings of the IEEE, vol. 73, no. 9, pp. 1349-1387, 1985.

[Ren70] A. Renyi, *Probability Theory*, North-Holland Publishing Company, Amsterdam, 1970.

[Ren76] A. Renyi, “Some Fundamental Questions of Information Theory*”.* *Selected Papers of Alfred Renyi*, vol. 2, pp. 526-552, Akademia Kiado, Budapest, 1976.

[Ren87] A. Renyi, *A Diary on Information Theory*, Wiley, New York, 1987.

[Rip96] B. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, New York, 1996.

[Rub81] R.Y. Rubinstein, *Simulation and the Monte Carlo Method*, John Wiley & Sons, New York, 1981.

[Rum86] D. Rumelhart, G. Hinton, R. Williams, “Learning Internal Representations by Error Back-propagation,” Nature, vol. 323, pp. 533-536, 1986.

[Sah97] P. Sahoo, C. Wilkins, and J. Yeager, "Threshold Selection Using Renyi's Entropy," Pattern Recognition, vol. 30, pp. 71-84, 1997.

[San02a] I. Santamaria, D. Erdogmus, J.C. Principe, "Entropy Minimization for Supervised Digital Communications Channel Equalization", to appear in IEEE Transactions on Signal Processing, 2002.

[San02b] I. Santamaria, C. Pantaleon, L. Vielva, J.C. Principe, “A Fast Algorithm for Adaptive Blind Equalization Using Order-α Renyi’s Entropy,” accepted to ICASSP’02, Orlando, FL, 2002.

[Sha48] C.E. Shannon, “A Mathematical Theory of Communications”, Bell Systems Technical Journal, vol. 27, pp. 379-423, pp. 623-656, 1948.

[Sha64] C.E. Shannon, W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, 1964.

[Sty90] M.A. Styblinski, T.S. Tang, “Experiments in Nonconvex Optimization: Stochastic Approximation with Function Smoothing and Simulated Annealing,” Neural Networks, vol. 3, pp. 467-483, 1990.

[Tak81] F. Takens, “On Numerical Determination of the Dimension of an Attractor,” in *Dynamical Systems and Turbulance*, (D. Rand, L.S. Young eds.), Warwick 1980, *Lecture Notes in Mathematics*, vol. 898, pp. 366-381, Springer-Verlag, Berlin, 1981.

[Tar68] F.P. Tarasenko, “On the Evaluation of an Unknown Probability Density Function, the Direct Estimation of the Entropy from Independent Observations of a Continuous Random Variable, and the Distribution-Free Entropy Test of Goodness-of-fit,” Proceedings of IEEE, vol. 56, pp. 2052-2053, 1968.

[Tit00] M.R. Titchener, “A Measure of Information,” Proceedings of the Data Compression Conference, pp. 353-362, 2000.

[Tor00] K. Torkkola, "Visualizing Class Structure in Data Using Mutual Information," Proceedings of NNSP X, pp. 376-385, Sydney, Australia, 2000.

[Tre83] J.R. Treichler, B.G. Agee, "A New Approach to Multipath Correction of Constant Modulus Signals," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 31, no. 4, pp. 349-372, 1983.

[Tsy94] A.B. Tsybakov, E.C. van der Meulen, "Root-n Consistent Estimators of Entropy for Densities with Unbounded Support," Scandinavian Journal of Statistics, vol. 23, pp. 75-83, 1994.

[Vap95] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, New York, 1995.

[Ver78] A.C.G. Verdugo Lazo, P.N. Rathie, "On the Entropy of Continuous Probability Distributions," IEEE Transactions on Information Theory, vol. 24, pp. 120-122, 1978.

[Vio95] P. Viola, N. Schraudolph, T. Sejnowski, "Empirical Entropy Manipulation for Real-World Problems", Proceedings of NIPS'95, pp. 851-857, 1995.

[Vog88] T.P. Vogl, J.K. Mangis, A.K. Rigler, W.T. Zink, D.L.Alkon, "Accelerating the Convergence of Back-Propagation Method," Biological Cybernetics, vol. 59, pp. 257-263, 1988.

[Wan99] C. Wang, J.C. Principe, "Training Neural Networks with Additive Noise in the Desired Signal," IEEE Transactions on Neural Networks, vol. 10, no. 6, pp. 1511-1517, 1999.

[Wat87] R.L. Watrous, "Learning Algorithms for Connectionist Networks: Applied Gradient Methods of Nonlinear Optimization," Proceedings of ICNN'87, vol. 2, pp. 619-627, San Diego, CA, 1987.

[Web88] A.R. Webb, D. Lowe, M.D. Bedworth, "A Comparison of Non-Linear Optimisation Strategies for Feed-Forward Adaptive Layered Networks," RSRE Memorandum 4157, Royal Signals and Radar Establishment, Malvern, UK, 1988.

[Wei84] A.S. Weigend, N.A. Gershenfeld, *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, Reading, MA, 1984.

[Wid85] B. Widrow, S.D. Stearns, *Adaptive Signal Processing*, Prentice Hall, New Jersey, 1985.

[Wie49] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications*, MIT Press, Cambridge, MA, 1949.

[Wu99] H.-C. Wu, *Blind Source Separation Using Information Measures in the Time and Frequency Domains*, PhD Dissertation, University of Florida, 1999.

[Xiq99] L. Xiqun, C. Chen, "A New Hybrid Recurrent Neural Network," Proceedings of ISCAS'99, vol. 5, pp. 616-618, 1999.

[Xu99] D. Xu, *Energy, Entropy and Information Potential for Neural Computation*, PhD Dissertation, University of Florida, 1999.

[Xu98] D. Xu, J.C. Principe, J. Fisher, H. Wu, "A Novel Measure for Independent Component Analysis (ICA)," in Proceedings of ICASSP'98, vol. 2, pp. 1161-1164, 1998.

[Yam97] Y.F. Yam, T.W.S. Chow, "A New Method in Determining the Initial Weights of Feedforward Neural Networks," Neurocomputing, vol. 16, no. 1, pp. 23-32, 1997.

[Yan97] H. Yang and S. Amari, "Adaptive Online Learning Algorithms for Blind Separation: Maximum Entropy and Minimum Mutual Information," Neural Computation, vol. 9, pp. 1457-1482, 1997.

BIOGRAPHICAL SKETCH

Deniz Erdogmus was born in Sivas, Turkey, on April 18, 1976. He received his B.S. in electrical and electronics engineering and B.S. in mathematics in 1997, and his M.S. in electrical and electronics engineering, with emphasis on systems and control, in 1999, all from the Middle East Technical University, Ankara, Turkey. From 1997 to 1999, he worked as a research engineer specializing on guidance, navigation, and flight control algorithms, at the Defense Industries Research and Development Institute (SAGE) of The Scientific and Technical Research Council of Turkey (TUBITAK). Since 1999, he has been working toward his Ph.D. at the Electrical and Computer Engineering Department at the University of Florida, under the supervision of Jose C. Principe. His current research interests broadly include information theoretic learning, adaptive systems, and signal processing. He is a member of IEEE, Tau Beta Pi and Eta Kappa Nu.