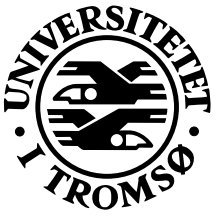
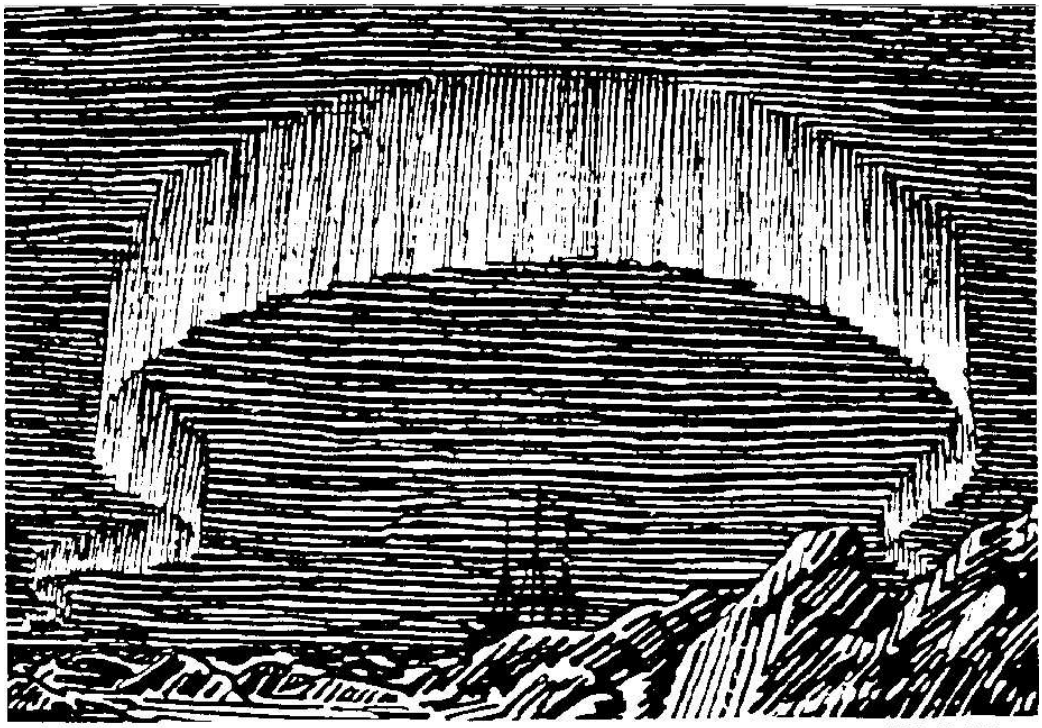


A Dissertation for the Degree of Doctor Scientiarum

An Information Theoretic Approach to Machine Learning

Robert Jenssen

May 2005

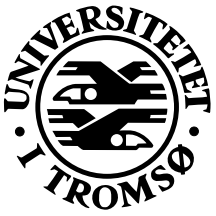


Original by F. Nansen

FACULTY OF SCIENCE

Department of Physics

University of Tromsø, NO-9037 Tromsø, Norway, telephone: +47 77 64 51 50, fax no: +47 77 64 55 80

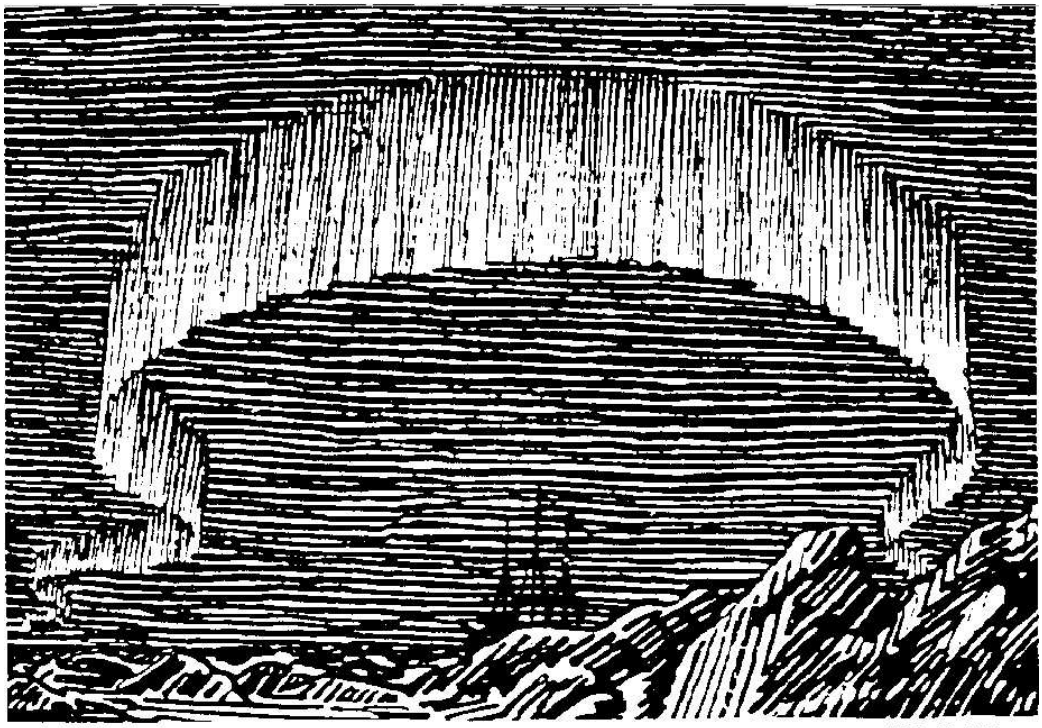


A Dissertation for the Degree of Doctor Scientiarum

An Information Theoretic Approach to Machine Learning

Robert Jenssen

May 2005



Original by F. Nansen

FACULTY OF SCIENCE

Department of Physics

University of Tromsø, NO-9037 Tromsø, Norway, telephone: +47 77 64 51 50, fax no: +47 77 64 55 80

To Guro

Abstract

In this thesis, theory and applications of machine learning systems based on information theoretic criteria as performance measures are studied.

A new clustering algorithm based on maximizing the Cauchy-Schwarz (CS) divergence measure between probability density functions (pdfs) is proposed. The CS divergence is estimated non-parametrically using the Parzen window technique for density estimation. The problem domain is transformed from discrete $0/1$ cluster membership values to continuous membership values. A constrained gradient descent maximization algorithm is implemented. The gradients are stochastically approximated to reduce computational complexity, making the algorithm more practical. Parzen window annealing is incorporated into the algorithm to help avoid convergence to a local maximum. The clustering results obtained on synthetic and real data are encouraging.

The Parzen window-based estimator for the CS divergence is shown to have a dual expression as a measure of the cosine of the angle between cluster mean vectors in a feature space determined by the eigenspectrum of a Mercer kernel matrix. A spectral clustering algorithm is derived and implemented in feature spaces defined by the spectrum of the affinity matrix and the Laplacian matrix, respectively. Using tools from statistics for Parzen window-size selection, the new spectral algorithm operates in a fully automatic mode with respect to the width of the Mercer kernel. A connection to the graph cut is also provided. The performance of the new algorithm is quite promising.

It is further shown that Parzen window-based estimators for Renyi's quadratic entropy and an integrated squared error (ISE) pdf divergence can also be expressed as functions of mean vectors in a Mercer kernel feature space. A new classification rule based on the ISE is proposed an studied theoretically. It is shown to be a hyperplane classifier in the kernel feature space, and a special case of the support vector machine (SVM). By introducing weighted Parzen window density estimators, an information theoretic interpretation of the SVM is provided.

An application of independent component analysis (ICA) is studied. Image basis functions are created by presenting textured training data to the FastICA algorithm. These texture basis functions are shown to capture the properties of the texture, and are used as a filter bank for generating energy features for segmentation of textured images. The ICA filter bank yields similar or better results than the Gabor filter bank.

Acknowledgments

I would like to express my sincere gratitude to my supervisor Professor Torbjørn Eltoft for all his guidance and help over the course of my doctoral studies. This work is also his work. Torbjørn has a great collaborative and relaxed style of supervising, which I appreciate very much. I hope we can conduct research together also in the continuation of my career.

I had the great pleasure of visiting the Computational NeuroEngineering Laboratory (CNEL) at the University of Florida for the academic year 2002/2003 and March/April 2004. I sincerely thank the Director of CNEL, Distinguished Professor Jose C. Principe, for hosting me. I found that Dr. Principe also has a very positive supervising style which encourages independent thinking, and I greatly appreciate his help and our fruitful collaboration. My stay at CNEL heavily influenced my research interests and the contents of this thesis. Especially, all the discussions and collaboration with Assistant Professor Deniz Erdogmus is sincerely acknowledged. I also extend my thanks to my office-mate at CNEL, Dr. Kenneth E. Hild II, for all collaboration and help. I would further like to thank Ellie for making me feel at home and for her wonderful sense of humor, Liu for many interesting conversations and to all the rest at CNEL for being a such a nice group of people.

I would like to acknowledge both former and current fellow students and employees at the Department of Physics at the University of Tromsø for making it a very nice place to study and work. Especially, I thank Associate Professor Jarle A. Johansen and Dr. Tor Arne Øigård for proofreading parts of the thesis and Dr. Arnt-Børre Salberg for many useful discussions.

I would like to acknowledge the University of Tromsø for granting the scholarships which made my research stays in Florida possible, and Professor Alfred Hanssen for financial travel support. I am also very thankful to the Department of Physics for employing me for the academic year 2005/2006.

I would like to thank Professor Klaus-Robert Müller and Professor Erkki Oja for taking the time to serve on my committee.

Finally, I would like to take this opportunity to acknowledge the value of having a great family. Thank you Britt, Bjørnar, Kristine, Robert D., Marte Amalie and Kristoffer for that. In particular, I deeply thank my lovely Guro for her patience and for the good life we share together.

Robert Jenssen,
Tromsø, Norway, March 2005.

Contents

Chapter 1

Introduction

In the first part of this chapter, we motivate an information theoretic approach to machine learning. In the second part, we provide an extended summary of the research reported in this thesis.

1.1 Motivation and Brief Review

The concept of *information theory* has traditionally been associated with the communications area, where it has had a tremendous impact in the design of efficient and reliable communication systems [Shannon and Weaver, 1949, Cover and Thomas, 1991, Fano, 1961]. Shannon [1948] laid down the foundations of information theory, by defining a quantitative measure of the uncertainty, or information, associated with the outcome of a stochastic experiment described by a probability distribution. This measure was named *entropy*, and made it possible to answer key questions in communications, like e.g. questions related to the maximal amount of information that can be transferred through a particular channel, and to the design of optimal codes for the data compression. Even though information theory is mainly associated with communications, it is a deep mathematical theory, which has had significant importance in fields such as probability theory, statistical inference, computer science, mathematics, physics, economics, biology and chemistry [Verdu, 2000].

In recent years, information theory has had an increasing impact on the important issue of extracting information directly from data, that is, *learning-from-examples*. The learning-from-examples scenario starts with a data set that globally conveys information about a real-world event, and the goal is to capture the information in the parameters of a learning machine [Principe et al., 2000a]. This *machine learning* problem often involves an adaptation process, where the parameters of the learning system are adjustable in such a way that performance improves through repeated presentation of exemplars to the system. The performance measure that is adopted will determine the type of information which can be extracted from the data.

Traditionally, the workhorse of adaptive systems has been the mean squared error (MSE) criterion [Haykin, 2000, Principe et al., 2004], and for good reasons. Since early research mainly was concentrated on linear adaptive systems, the adoption of such second-order statistics optimality measures resulted in quadratic performance surfaces, for which the analytical expression of the optimal solution could easily be obtained [Widrow and Stearns, 1985, Haykin, 2002]. The MSE criterion also carried over to non-linear adaptive systems and

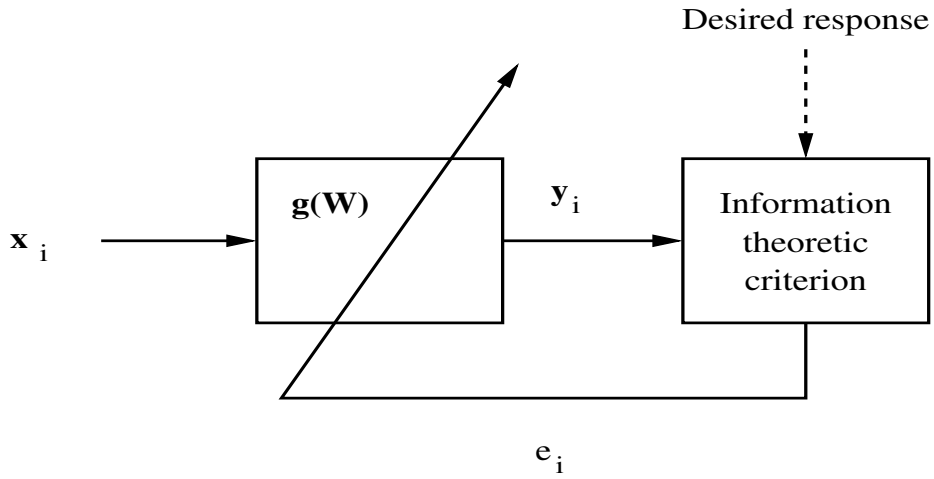


Figure 1.1: *In information theoretic machine learning, the output is given by* $\mathbf{y}_i = \mathbf{g}(\mathbf{W})\mathbf{x}_i$*, where* $\mathbf{x}_i$ *is the example presented to the system at iteration* i*. The function* $\mathbf{g}(\mathbf{W})$ *represents a possibly non-linear data transformation, which depends on the parameters given by* $\mathbf{W}$*. The goal is to train the system to perform a specific task, according to an information theoretic criterion. The criterion is evaluated at each iteration, and outputs a correction term* e_i *which is fed back to the system to guide the adjustment of the system parameters. The system may receive external input in the form of a desired response, in which the system operates in a supervised mode.*

neural networks. The backpropagation algorithm used for training multilayer perceptrons is an example of this.

However, for some problems second order statistics is insufficient to extract the structure of the data. For example, in the context of visual feature extraction, Barlow [1989] suggested that the sensory system uses a redundancy reduction process, in which the activation of feature detectors is supposed to be as statistically independent as possible. Second order statistics and statistical independence are only equivalent under the Gaussian assumption. Therefore entropy was proposed as a learning criterion [Barlow et al., 1989]. This work set the scene for several related feature-learning algorithms [Atick, 1992, Field, 1994, Intrator, 1992, Olshausen and Field, 1996].

Recently, many machine learning problems have been encountered which necessitate the use of cost functions which can capture higher order statistical properties of the data. Examples of such problems are blind source separation and independent component analysis, blind equalization and deconvolution, subspace projections, dimensionality reduction, feature extraction, classification and clustering. Such cost functions are provided by information theoretic criteria. Information theoretic criteria do capture all the data statistics, since they are functions of probability densities. In section 2.5 and 2.6, we will discuss the application of information theoretic criteria in clustering and independent component analysis, respectively.

Principe et al. [2000a] introduced the term *information theoretic learning* (ITL) as a general framework for machine learning and adaptive systems training using information theoretic criteria. Figure 1.1 shows a typical ITL system, which includes both the supervised and unsupervised learning schemes. We refer to the recent special issue published by the IEEE Transactions on Neural Networks for an overview of the state-of-the art approaches to ITL,

broadly defined [Schraudolph, 2004, Hulle, 2004, Morejon and Principe, 2004, Wang et al., 2004, Choi and Lee, 2004, Sanchez-Montanes and Corbacho, 2004, Cruces-Alvarez et al., 2004, Iwata et al., 2004, Rutkowski, 2004, Honkela and Valpola, 2004].

Specifically, Principe et al. [2000a] argued that one should make as few assumptions as possible about the structure of the probability density functions (pdfs) in question. Hence, Parzen windowing [Parzen, 1962, Devroye, 1989, Silverman, 1986, Scott, 1992, Wand and Jones, 1995] was proposed as the appropriate density estimation technique, since this method makes no such assumptions. Viola et al. [1995] had already proposed to approximate Shannon-based measures using sample means, integrated with Parzen windowing [Viola and Wells, 1997]. Principe et al. [2000a] went a step further, by introducing a series of information theoretic quantities which can be estimated without the sample mean approximation [Xu, 1999, Principe et al., 2000b]. This property may be important, since the sample mean approximation may not hold very well for small sample sizes. The proposed measures were based on generalizations of the Shannon entropy due to Renyi [1976b,a], and include Renyi's quadratic entropy, a divergence measure between pdfs based on the Cauchy-Schwarz (CS) inequality, and an integrated squared error divergence measure. Since these measures all include quantities which are expressed as integrals over products and squares of densities, we will refer to them as quadratic information measures. Information theoretic learning based on the quadratic information measures and Parzen windowing has been applied with great success by Principe and co-workers on several supervised and unsupervised learning problems [Erdogmus et al., 2005, Hild et al., 2005a,c,b, Lazaro et al., 2005, Erdogmus et al., 2004b,a, Sindhwani et al., 2004, Erdogmus and Principe, 2003, Erdogmus et al., 2002, Hild, 2003, Erdogmus, 2002, Santamaria et al., 2002, Erdogmus and Principe, 2002b,a, Erdogmus et al., 2002, Principe et al., 2000b]. These methods represent a very powerful approach to machine learning and adaptive systems training. A major part of this dissertation is based on this framework for information theoretic learning.

However, information theoretic learning is not the only scheme which has been developed and applied in machine learning over the last few years. Several other approaches exist [Jain et al., 2000, Kulkarni et al., 2000]. We will briefly mention two important learning schemes which have been developed recently, namely the Mercer kernel-based and the so-called spectral methods. The work conducted in this thesis is also related to these methods.

Perhaps the most well-known of these learning schemes is the Mercer kernel-based learning algorithms [Shawe-Taylor and Cristianini, 2004, Müller et al., 2001, Perez-Cruz and Bousquet, 2004, Schölkopf and Smola, 2002], of which support vector machines [Cortez and Vapnik, 1995, Vapnik, 1995, Cristianini and Shawe-Taylor, 2000, Burges, 1998, Hastie et al., 2004], kernel principal component analysis [Schölkopf et al., 1998], and kernel Fisher discriminant analysis [Mika et al., 1999, Roth and Steinhage, 2000] are examples. Research on Mercer kernel-based methods have been dominating in machine learning and pattern recognition over the last decade. The common property of these methods is that they are linear in nature and can be expressed solely in terms of inner-products. However, they may be applied to *non-linear* problems using the so-called "kernel trick". The kernel trick defines a technique for computing inner-products in a potentially infinite-dimensional *kernel feature space*, using so-called Mercer kernels. Mercer kernel-based methods have been applied successfully in applications like pattern and object recognition [LeCun et al., 1995], time series prediction [Müller et al., 1997] and DNA and protein analysis [Zien et al., 2000], to name a few.

Yet another popular family of machine learning algorithms has emerged recently. The common property of these algorithms is that they are based on an *eigendecomposition* of

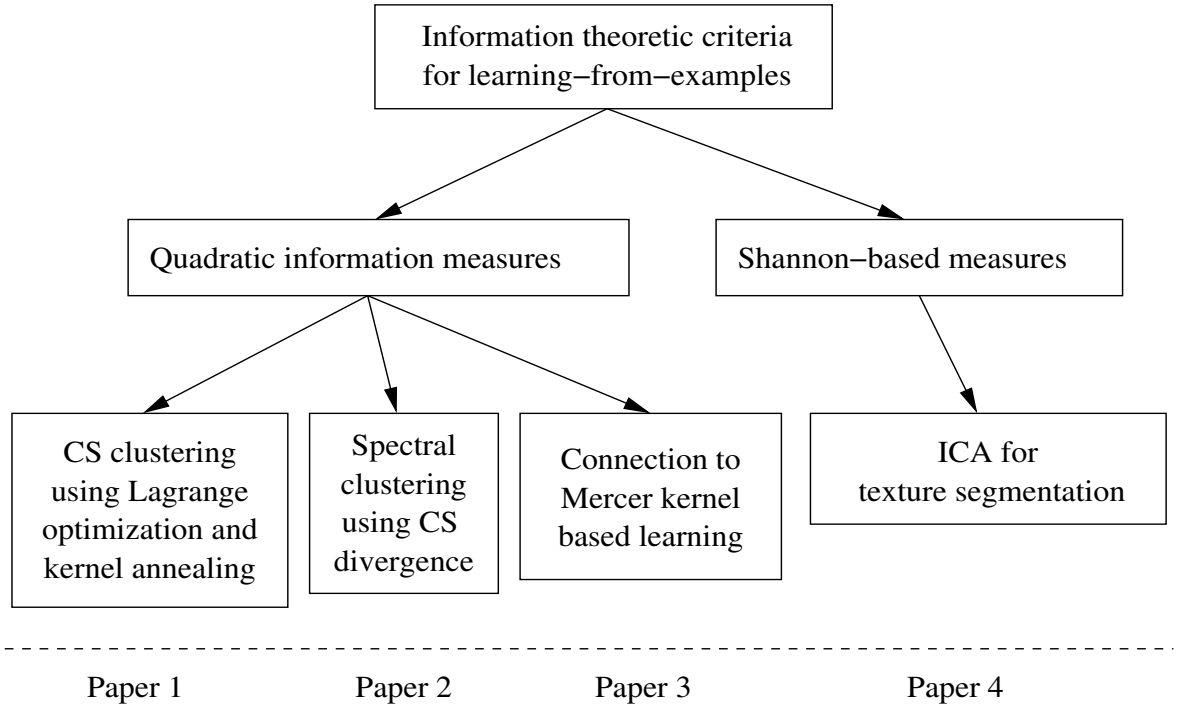


Figure 1.2: *Graphical illustration of the connections between the papers which constitute this thesis.*

an affinity matrix. The resulting eigenvalues and eigenvectors are key ingredients for performing tasks such as lower-dimensional embedding of the data on a non-linear manifold [Roweis and Saul, 2000, Tenenbaum et al., 2000, Belkin and Niyogi, 2003, Brand, 2004] or clustering [Ng et al., 2002, Weiss, 1999, Dhillon et al., 2004, Ding and He, 2004, Verma and Meila, 2003, Zha et al., 2002]. See also [Bengio et al., 2003, 2004]. Since the eigenvalues of such an affinity matrix are known as the *spectrum* of the matrix, these methods are referred to as spectral algorithms. For example, Google's Pagerank algorithm [Brin and Page, 1998] is based on related eigenvector calculations [Ng et al., 2001].

1.2 Extended Summary

This thesis considers both theoretical and application aspects of information theoretic learning, broadly defined. The research is presented in four papers, contained in chapters 3 to 6. Figure 1.2 illustrates how the different papers are connected. The illustration indicates two main directions of research in information theoretic learning methods. One branch concerns research based on the quadratic information measures and Parzen windowing. Papers 1-3 belong to this branch, and they are also internally related. The other branch refers to research based on Shannon-based information theory. Paper 4 belongs to this category.

Two appendices have been included. Appendix A discusses a connection between Parzen window-based estimators for quadratic information measures and graph theory. In Appendix B we have attached two conference papers. These show two variations of Cauchy-Schwarz-based clustering algorithms. Cauchy-Schwarz-based clustering algorithms are the topic of paper 1 (chapter 3) and paper 2 (chapter 4).

As a gesture to the reader unfamiliar with information theory, we provide in chapter 2 a brief overview of the information theoretic measures which are important to this dissertation. Clustering theory and independent component analysis are also briefly discussed.

Paper 1

R. Jenssen, D. Erdogmus, K. E. Hild II, J. C. Principe and T. Eltoft, "**Non-Parametric Clustering by Maximizing the Cauchy-Schwarz PDF Divergence**," unpublished manuscript draft.

In this paper, we develop a practical algorithm for data clustering based on the Cauchy-Schwarz divergence measure between probability density functions. The goal is to assign cluster memberships to the data samples such that the CS divergence between the estimated cluster pdfs is maximized. The cluster pdfs are themselves estimated from the data using the Parzen window technique.

The approach taken, is to transform the problem domain from discrete $0/1$ cluster membership values to continuous membership values, which are restricted to lie in the interval $[0, 1]$. This transformation, in combination with the Parzen window estimation, makes it possible to compute gradients of the CS clustering cost function with respect to the membership values. Hence techniques from differential calculus can be employed in the adjustment of the memberships. The actual maximization is carried out by an iterative Lagrange multiplier optimization technique that implements a constrained gradient descent search, with built-in variable step-sizes for each coordinate direction (i.e. no need for manually selected step size parameters). The gradients are expressed in terms of pairwise distances between all data points. To reduce complexity, the gradients are stochastically approximated by randomly selecting a small subset of the data for gradient computation. The resulting algorithm has a computational complexity of $O(MN)$, where $M \ll N$ is the number of randomly selected data points used in the stochastic approximation, and N is the number of data patterns to be clustered. The algorithm is independent of the order of data presentation. After convergence of the algorithm, the membership values are discretized such that a partitioning of the data set is obtained.

An added advantage of using Parzen windowing in the CS maximization is that the risk of convergence to a local optimum of the cost function can be reduced by allowing the kernel size to be annealed over a range of values around the optimal value. The optimal value is determined automatically from the data set. We show that the new clustering algorithm is capable of clustering irregularly shaped synthetic data sets, as well as other real data sets.

This work is partially based on the papers [Jenssen et al., 2003a,b,c].

Paper 2

R. Jenssen, D. Erdogmus, J. C. Principe and T. Eltoft, "**Spectral Clustering based on Information Theory and Parzen Windowing**," Journal of Machine Learning Research, resubmitted.

In this paper, we propose a new *spectral* clustering algorithm that maximizes the information theoretic Cauchy-Schwarz divergence measure between probability density functions. The clustering algorithm is based on the *generalized information cut*, which is a graph theoretic, Parzen window-based estimator for the CS divergence. This clustering cost function

is shown to measure the cosine of the angle between the cluster mean vectors in the feature space determined by the eigenvalues (the spectrum) and eigenvectors of a Mercer kernel matrix.

The algorithm starts by robustly initializing the mean vectors in the kernel space. The cosine of the angle between each feature space data point and the mean vectors are computed, and the data points are assigned to the cluster for which the cosine is maximum. Thereafter the mean vectors are recomputed. This iterative procedure is repeated until convergence.

For computational purposes, the dimensionality of the feature space is reduced such that it equals the number of clusters. The number of clusters is estimated based on the eigendecomposition of the kernel matrix. The construction of the kernel matrix is automated, at least for data sets of low to moderate dimensionality, based on Parzen window selection procedures.

The Mercer kernel matrix is determined both by the Parzen window and a non-negative weighting function. We examine two different weighting functions, corresponding to the affinity matrix and the Laplacian matrix, respectively. Clustering experiments, using several different data sets, are conducted to assess the performance of this clustering approach. The results are quite promising.

This work is partially based on the papers [Jenssen et al., 2005, 2003d, 2004b].

Paper 3

R. Jenssen, D. Erdogmus, J. C. Principe and T. Eltoft, "**Some Equivalences Between Kernel Methods and Information Theoretic Methods**," Journal of VLSI Signal Processing, special issue on machine learning for signal processing, submitted.

In this paper, some equivalences between Mercer kernel methods and information theoretic measures are discussed. It is shown that Parzen window-based estimators for the quadratic information measures are also measures in the Mercer kernel-based feature space. The *information potential* (Renyi quadratic entropy) is shown to measure the squared norm of the mean vector of the Mercer kernel-transformed data set. The *information cut* (Cauchy-Schwarz divergence) is shown to measure the cosine of the angle between cluster mean vectors in the Mercer kernel feature space. The integrated squared error (ISE) divergence is shown to measure the squared norm of the difference vector between feature space cluster mean vectors.

Since the Mercer kernel is shown to be directly related to the Parzen window, it is suggested that the Mercer kernel-size may be determined by optimal Parzen window-size selection procedures.

A new classification rule based on the ISE measure is proposed and studied theoretically both in the input space and in the Mercer kernel feature space. In the input space it is based on Parzen window density estimates and contains the Bayes classifier as a special case. In the Mercer kernel feature space it corresponds to a hyperplane classifier. The hyperplane is determined by the feature space cluster mean vectors. It is shown that the ISE classifier is a special case of the support vector machine (SVM). By introducing weighted Parzen window density estimators, an information theoretic interpretation of the SVM is provided.

This work is partially based on the paper [Jenssen et al., 2004c].

Paper 4

R. Jenssen and T. Eltoft, "**Independent Component Analysis for Texture Segmentation**," Pattern Recognition, 36(10): 2301-2315, 2003.

In this paper, independent component analysis (ICA) of textured images is presented as a computational technique for creating a new data dependent filter bank for texture segmentation. The underlying assumption is that ICA applied to images produces basis images which capture the inherent properties of the training data, that is, the data presented to the ICA algorithm. Hence, the filter bank consists of ICA basis images created using textured training data.

The majority of the resulting basis images are shown to be localized in spatial frequency and orientation, that is they seem to exhibit a kind of sinusoidal wave structure of a specific spatial frequency and to be oriented in a specific direction. The new filters are similar to Gabor filters, but seem to be richer in the sense that their frequency responses may be more complex. Hence, they seem to capture the inherent features of the textured training data.

A composite textured image is convolved with each of the ICA filters. Regions in the input image which are tuned to a filter tend to produce high energy in the corresponding regions of the filtered image, while the opposite effect is observed for regions which are not tuned to the filter. These properties are used to generate energy feature vectors corresponding to each pixel in the input image. These feature vectors are clustered. Each cluster corresponds to a region in the image, which yields the final segmentation.

Our experiments using multi-textured input images show that the ICA filter bank yields similar or better segmentation results than the Gabor filter bank.

This work is partially based on the paper [Jenssen and Eltoft, 2003].

Chapter 2

Information Theoretic Criteria

In this chapter we define some concepts pertaining to the information content associated with the outcome of a stochastic experiment. We also define some quantities which measure the divergence, or "distance" between probability density functions. We start by discussing the basic measures developed by Shannon [1948]. Thereafter, we will consider generalizations of these quantities due to Renyi [1976b,a], and also consider some pdf divergence measures proposed recently by Principe et al. [2000a]. These measures are referred to as information theoretic criteria. The material is treated in some detail, since the measures provide the theoretical backbone of the thesis. We also review some density estimation methods, which are needed to evaluate information theoretic quantities. We conclude the chapter by discussing clustering and independent component analysis, two topics that exemplify important applications in this area of research.

Definitions

A discrete stochastic variable $\mathbf{X}$ is associated with a triple $(\mathbf{x}, \mathcal{A}_{\mathbf{X}}, \mathcal{P}_{\mathbf{X}})$, where the outcome $\mathbf{x}$ is the value of the stochastic variable which takes on a set of possible values $\mathcal{A}_{\mathbf{X}} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N\}$. These have probabilities (distribution) $\mathcal{P}_{\mathbf{X}} = \{p_1, p_2, \ldots, p_N\}$, where $P(\mathbf{x} = \mathbf{a}_i) = p_i$, $p_i \geq 0$ and $\sum_{\mathbf{a}_i \in \mathcal{A}_{\mathbf{X}}} P(\mathbf{x} = \mathbf{a}_i) = 1$.

A continuous stochastic variable $\mathbf{X}$ is associated with a probability density function $f_{\mathbf{X}}(\mathbf{x})$, where the outcome $\mathbf{x}$ is the value of the stochastic variable. The pdf is defined as the derivative of the cumulative distribution function (cdf), defined as $P(\mathbf{x} \leq \mathbf{x}_0) = F_{\mathbf{X}}(\mathbf{x}_0)$, where $0 \leq F_{\mathbf{X}}(\mathbf{x}) \leq 1$. Hence, $f_{\mathbf{X}}(\mathbf{x}_0) = \frac{\partial}{\partial \mathbf{x}} F_{\mathbf{X}}(\mathbf{x})\big|_{\mathbf{x}=\mathbf{x}_0}$, and $\int_{-\infty}^{\infty} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = 1$. For notational simplicity, we use $f(\mathbf{x}) = f_{\mathbf{X}}(\mathbf{x})$.

Mathematically, the random variables $\mathbf{X}_1, \ldots, \mathbf{X}_N$, are said to be *statistically independent* if and only if $f(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \prod_{i=1}^{N} f(\mathbf{x}_i)$. In words, the joint density $f(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ must factorize into the product of the marginal densities $f(\mathbf{x}_i),\ i = 1 \ldots, N$.

2.1 Shannon Entropy and Kullback-Leibler Divergence

Consider an experiment where there is some uncertainty as to the outcome of the experiment. Assume that the possible outcomes of the experiment are described by a probability

distribution. Hence, the experiment has some "uncertainty" associated with it. Shannon [1948] was the first to define a quantitative measure, $H = H_N(p_1, \ldots, p_N)$, of this uncertainty, satisfying the following set of basic postulates [Shannon, 1948, Renyi, 1976a]

1. $H_N(p_1, \ldots, p_N)$ is a symmetric function of its variables.
2. $H_N(p_1, \ldots, p_N)$ is a continuous function of $p_1, \ldots, p_N$.
3. $H_N\left(\frac{1}{N}, \ldots, \frac{1}{N}\right)$ attains the maximum value.
4. $H_{N+1}(tp_1, (1-t)p_1, p_2, \ldots, p_N) = H_N(p_1, \ldots, p_N) + p_1 H_2(t, 1-t)$ for any distribution $\mathcal{P}_{\mathbf{X}}$ and $0 \leq t \leq 1$.

The fourth property is known as the recursive property [Kapur, 1994]. It basically states that if a choice be broken down into two successive choices, the new H_{N+1} can be computed from the original H_N and $H_2(t, 1-t)$.

Shannon Entropy

Shannon showed that the only H satisfying the above assumptions is given by [Shannon, 1948, Shannon and Weaver, 1949]

$$H_N(p_1, \ldots, p_N) = H_N(\mathcal{P}_{\mathbf{X}}) = -\sum_{p_i \in \mathcal{P}_{\mathbf{X}}} p_i \log_b p_i, \quad [1] \tag{2.1}$$

with the convention that $0 \log_b 0 = 0$. This measure was named the *entropy*, because it was recognized as the same expression used to define entropy in statistical mechanics. If $b = 2$, the unit of the entropy is $bits$. For $b = e$, the unit is $nats$. The base only changes the measurement scale, and is not important in this exposition. Hence, we do not specify b in the following discussion. It is also common to denote entropy by $H(\mathbf{X})$, in which case it should be noted that the $\mathbf{X}$ in $H(\mathbf{X})$ is not an argument of a function, but rather a label for a random variable.

It can be seen that the Shannon entropy depends on the quantity $I(p_i) = -\log p_i$. The measure $I(p) = -\log p$ was proposed by Hartley [1928] as a measure of the information received by learning that a single event of probability p took place. Hence, the Shannon entropy is a weighted average of the informations $I(p_i)$.

When an experiment is performed and we know the actual outcome, the uncertainty is removed. As such the information provided by the experiment can be regarded as equal to the amount of uncertainty removed by it. Thus a measure of entropy can also be regarded as the measure of information provided by the realization of a probability distribution [Kapur, 1989].

Several other properties of the Shannon entropy can be derived from the four basic properties [Shannon, 1948, Kapur, 1989]

(i) It does not change if an impossible outcome is added to the probability scheme, i.e.

$$H_{N+1}(p_1, \ldots, p_N, 0) = H_N(p_1, \ldots, p_N). \tag{2.2}$$

[1] Actually, $H = -K \sum_{p_i \in \mathcal{P}_{\mathbf{X}}} p_i \log_b p_i$ satisfies the assumptions, K being a positive constant which merely amounts to a choice of a unit of measure.

(ii) It vanishes when one of the outcomes is certain to happen, so that

$$H_N(p_1, \ldots, p_N) = 0, \quad p_i = 1,\ p_j = 0,\ j \neq i,\ i = 1, \ldots, N. \tag{2.3}$$

(iii) The maximum of H_N (see property 3. above) increases as N increases.

(iv) For two independent probability distributions $\mathcal{P}_{\mathbf{X}} = \{p_1, \ldots, p_N\}$ and $\mathcal{Q}_{\mathbf{Y}} = \{q_1, \ldots, q_M\}$, $\sum_{i=1}^{N} p_i = 1$ and $\sum_{j=1}^{M} q_j = 1$, the uncertainty of the joint scheme $\mathcal{P}_{\mathbf{X}} \cup \mathcal{Q}_{\mathbf{Y}}$ is the sum of their uncertainties

$$H_{N+M}(\mathcal{P}_{\mathbf{X}} \cup \mathcal{Q}_{\mathbf{Y}}) = H_N(\mathcal{P}_{\mathbf{X}}) + H_M(\mathcal{Q}_{\mathbf{Y}}). \tag{2.4}$$

This will be referred to as the *additivity* property. It can be easily proved by evaluating the entropy of the joint event using $P(\mathbf{x} = \mathbf{a}_i, \mathbf{y} = \mathbf{b}_j) = p(i, j)$, and

$$H_{N+M}(\mathcal{P}_{\mathbf{X}} \cup \mathcal{Q}_{\mathbf{Y}}) = -\sum_{i=1}^{N} \sum_{j=1}^{M} p(i, j) \log p(i, j), \tag{2.5}$$

substituting $p(i, j)$ by $p_i q_j$. $H_{N+M}(\mathcal{P}_{\mathbf{X}} \cup \mathcal{Q}_{\mathbf{Y}})$ is often written as $H(\mathbf{X}, \mathbf{Y})$.

(v) Assume now that the two schemes are not necessarily independent, and $P(\mathbf{x} = \mathbf{a}_i, \mathbf{y} = \mathbf{b}_j) = p(i, j)$. Then, it can be shown [Shannon, 1948, Kapur, 1989] that

$$H_{N+M}(\mathcal{P}_{\mathbf{X}} \cup \mathcal{Q}_{\mathbf{Y}}) = H_N(\mathcal{P}_{\mathbf{X}}) + \sum_{i=1}^{N} p_i H_M(\mathcal{Q}_{\mathbf{Y}} | \mathbf{x} = \mathbf{a}_i), \tag{2.6}$$

where $\sum_{i=1}^{N} p_i H_M(\mathcal{Q}_{\mathbf{Y}} | \mathbf{x} = \mathbf{a}_i) = \sum_{i=1}^{N} \sum_{j=1}^{M} p(i, j) \log p(j|i)$ is often denoted $H(\mathbf{Y}|\mathbf{X})$. Hence, the measure of uncertainty for a joint probability scheme is the sum of the measure of uncertainty of one of the schemes and the mathematical expectation of the measure of uncertainty of the other, conditional on the realization of the other scheme.

Renyi [1976a] considered generalized probability distributions for which $w(\mathcal{P}_{\mathbf{X}}) = \sum_{i=1}^{N} p_i \leq 1$ and $w(\mathcal{Q}_{\mathbf{Y}}) = \sum_{j=1}^{M} q_j \leq 1$, and rewrote Eq. (2.6) as the following mean value

$$H_{N+M}(\mathcal{P}_{\mathbf{X}} \cup \mathcal{Q}_{\mathbf{Y}}) = \frac{w(\mathcal{P}_{\mathbf{X}}) H_N(\mathcal{P}_{\mathbf{X}}) + w(\mathcal{Q}_{\mathbf{Y}}) H_M(\mathcal{Q}_{\mathbf{Y}})}{w(\mathcal{P}_{\mathbf{X}}) + w(\mathcal{Q}_{\mathbf{Y}})}. \tag{2.7}$$

We will return to this expression shortly, when we define the Renyi entropy.

Example

A simple example illustrates some of the properties of the Shannon entropy. Assume that a stochastic variable $\mathbf{X}$ can take only two outcomes, where $p_1 = p$ and $p_2 = 1 - p_1 = 1 - p$. Hence [Shannon, 1948, Cover and Thomas, 1991]

$$H(p) = -p \log p - (1 - p) \log(1 - p). \tag{2.8}$$

Figure 2.1 shows the graph of the function $H(p)$. It can be seen that $H(p)$ equals 0 when $p = 0$ or 1. This makes sense, since when $p = 0$ or 1 the variable is not stochastic and there is no uncertainty. Similarly, the uncertainty is maximum when $p = \frac{1}{2}$, which corresponds to the maximum value of the entropy.

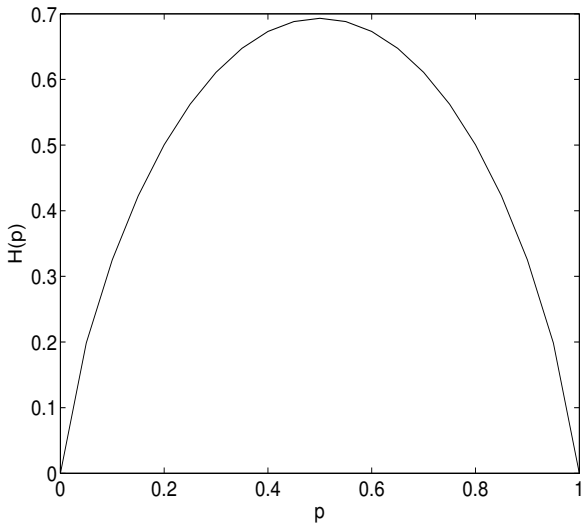


Figure 2.1: $H(p)$ *(nats) versus* p*.*

Shannon Differential Entropy

For a continuous stochastic variable, the counterpart of Eq. (2.1) is called the *differential entropy*, and is given by [Shannon and Weaver, 1949]

$$h(\mathbf{X}) = -\int f(\mathbf{x}) \log f(\mathbf{x}) d\mathbf{x}.^{2} \tag{2.9}$$

Note that $h(\mathbf{X})$ can also be written as an expected value, that is,

$$h(\mathbf{X}) = E_f \left\{ -\log f(\mathbf{x}) \right\}, \tag{2.10}$$

where $E_f\{\cdot\}$ denotes expectation over $f(\mathbf{x})$. As in the discrete case, the differential entropy depends only on the probability density of the stochastic variable.

Likewise, the joint and conditional differential entropies are defined as

$$\begin{aligned} h(\mathbf{X}, \mathbf{Y}) &= -\int f(\mathbf{x}, \mathbf{y}) \log f(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}, \\ h(\mathbf{Y}|\mathbf{X}) &= -\int f(\mathbf{x}, \mathbf{y}) \log f(\mathbf{y}|\mathbf{x}) d\mathbf{x} d\mathbf{y}, \end{aligned} \tag{2.11}$$

and the following hold, just as in the discrete case

(i) $h(\mathbf{X}, \mathbf{Y}) = h(\mathbf{X}) + h(\mathbf{Y}|\mathbf{X})$, where $h(\mathbf{Y}|\mathbf{X}) = h(\mathbf{Y})$ iff $\mathbf{X}$ and $\mathbf{Y}$ are independent.

(ii) $h(\mathbf{Y}) \geq h(\mathbf{Y}|\mathbf{X})$, i.e., the uncertainty of $\mathbf{Y}$ is never increased by the knowledge of $\mathbf{X}$.

[2]The definition should be used with caution, since one may construct stochastic variables for which a density function does not exist or for which the integral does not exist [Cover and Thomas, 1991]. Also, note that all integrals range from $-\infty$ to ∞ unless stated otherwise.

Differential entropy can also be interpreted as a measure of randomness, and it possesses most (but not all) of the same properties as the Shannon entropy for discrete variables. In particular, some properties of the Shannon differential entropy are

1. If $\mathbf{X}$ is limited to a certain volume v in its space, then $h(\mathbf{X})$ is maximum and equal to $\log v$ when $f(\mathbf{x})$ is constant $\frac{1}{v}$ (uniform density function) in the volume.

2. Differential entropy may be negative. Consider the uniform density discussed above. For $v < 1$, $\log v < 0$.

3. The normal distribution maximizes the entropy over all distributions with the same covariance. This property can be exploited in order to measure the non-Gaussianity of a stochastic variable. A function known as the negentropy can be defined as

$$J(\mathbf{X}) = h(\mathbf{X}_{\text{Gauss}}) - h(\mathbf{X}), \tag{2.12}$$

   where $\mathbf{X}_{\text{Gauss}}$ is a Gaussian stochastic variable having the same covariance matrix as $\mathbf{X}$.

4. The differential entropy is a measure which is relative to the coordinate system. Consider for example changing coordinates by a linear transformation, $\mathbf{Y} = \mathbf{M}\mathbf{X}$. In that case,

$$h(\mathbf{Y}) = h(\mathbf{X}) + \log |det(\mathbf{M})| , \tag{2.13}$$

   where $|det(\mathbf{M})|$ is the absolute value of the determinant of $\mathbf{M}$. Note that if $\mathbf{M}$ is a rotation matrix (orthogonal), $\log |det(\mathbf{M})| = 0$. That is, the Shannon differential entropy is invariant under rotations. It is also invariant to translations, i.e. $h(\mathbf{X} + \mathbf{c}) = h(\mathbf{X})$.

When working with continuous stochastic variables, it is quite common to refer to the differential entropy simply as entropy.

Kullback-Leibler Divergence

In a classic paper, Kullback and Leibler [1951] considered the statistical problem of discrimination, by considering a measure of the divergence or "distance" between statistical populations. They proposed such a measure, subsequently called the Kullback-Leibler divergence, or the relative entropy. The measure discriminates between two probability density functions $f(\mathbf{x})$ and $g(\mathbf{x})$ and is given by [Kullback and Leibler, 1951, Kullback, 1968]

$$\begin{aligned} D_{KL}\{f, g\} &= \int f(\mathbf{x}) \log \frac{f(\mathbf{x})}{g(\mathbf{x})} d\mathbf{x}, \\ &= E_f \left\{ \log \frac{f(\mathbf{x})}{g(\mathbf{x})} \right\}. \end{aligned} \tag{2.14}$$

The term "distance" is a misnomer because this measure is not a distance metric in the mathematical sense, since it is not symmetric, nor does it satisfy the triangle inequality [Cover and Thomas, 1991].

Some properties possessed by the Kullback-Leibler divergence are the following [Kullback and Leibler, 1951]

1. $D_{KL}\{f, g\} \geq 0, \ \forall f, g.$

2. $D_{KL}\{f, g\} = 0$, iff $f(\mathbf{x}) = g(\mathbf{x})$, $\forall \mathbf{x} \in R^d$.

3. $D_{KL}\{f, g\}$ is additive for independent random events.

Properties one and two can be easily proved using Jensen's inequality [Cover and Thomas, 1991]. Property number three is one of the a priori fundamental requirements for an information measure set down by Shannon. Let us assume that $\mathbf{X} = [X_1, X_2]^T$, such that $f(\mathbf{x}) = f(x_1)f(x_2)$ and $g(\mathbf{x}) = g(x_1)g(x_2)$ by independence. Then the additivity property states that

$$D_{KL}\{f(\mathbf{x}), g(\mathbf{x})\} = D_{KL}\{f(x_1), g(x_1)\} + D_{KL}\{f(x_2), g(x_2)\}. \tag{2.15}$$

The Kullback-Leibler divergence is also invariant under the following changes in the vector $\mathbf{x}$ [Haykin, 1999]

1. Permutation of the order in which the components are arranged.

2. Amplitude scaling.

3. Monotonic nonlinear transformation.

It can be seen that the Kullback-Leibler divergence is implicitly based on Shannon's entropy, since

$$D_{KL}\{f, g\} = -\int f(\mathbf{x}) \log g(\mathbf{x}) d\mathbf{x} - \left(-\int f(\mathbf{x}) \log f(\mathbf{x}) d\mathbf{x} \right). \tag{2.16}$$

We recognize $-\int f(\mathbf{x}) \log f(\mathbf{x}) d\mathbf{x}$ as Shannon's differential entropy with respect to $f(\mathbf{x})$. The quantity $-\int f(\mathbf{x}) \log g(\mathbf{x}) d\mathbf{x}$ can be interpreted as a "cross-entropy" between $f(\mathbf{x})$ and $g(\mathbf{x})$.

Mutual Information

The Kullback-Leibler divergence may be used as a tool to measure the independence between random variables. Let $\mathbf{X} = [X_1, X_2]^T$. If the random variables are stochastically independent, we have $f(\mathbf{x}) = f(x_1)f(x_2)$. The Kullback-Leibler divergence between $f(\mathbf{x})$ and $f(x_i)f(x_2)$ is called the mutual information between X_1 and X_2. That is,

$$\text{MI}(X_1, X_2) = \int f(\mathbf{x}) \log \frac{f(\mathbf{x})}{f(x_1)f(x_2)} dx_1 dx_2. \tag{2.17}$$

It is a measure of the information that one random variable has on the other random variable. The extension to the case that $\mathbf{X} = [X_1, \ldots, X_N]^T$ is straightforward. The mutual information is always non-negative since it is based on the Kullback-Leibler divergence. It can be seen that it is zero if and only if the variables are independent, and that it is symmetric.

The mutual information can be expressed in terms of Shannon entropies, as [Cover and Thomas, 1991]

$$\begin{aligned} \text{MI}(X_1, X_2) &= h(X_1) - h(X_1|X_2) = h(X_2) - h(X_2|X_1). \\ \text{MI}(X_1, X_2) &= h(X_1) + h(X_2) - h(\mathbf{X}). \end{aligned} \tag{2.18}$$

2.2 Renyi Entropy and Renyi Divergence

Renyi [1976a] further considered property (v) of the Shannon entropy, which he coined the *mean value property* of entropy. He was interested in what other quantity he would obtain by replacing the arithmetic mean in Eq. (2.7) with some other mean value. This led to the definition of the Renyi entropy of order α. A similar derivation led to the definition of the Renyi divergence, which relates to the Kullback-Leibler divergence in much the same way as the Renyi entropy and the Shannon entropy.

Renyi Entropy

In the general theory of means, the mean of the real numbers $x_1, \ldots, x_N$ with positive weighting $w_1, \ldots, w_N$ has the form [Renyi, 1976a]

$$\overline{x} = \phi^{-1}\left[\sum_{i=1}^{N} w_i \phi(x_i)\right], \tag{2.19}$$

where $\phi(x)$ is a Kolmogorov-Nagumo function, which is an arbitrary continuous and strictly monotonic function defined on the real numbers [Renyi, 1976b], and ϕ^{-1} denotes the inverse function of ϕ. This led Renyi to replace Eq. (2.7) in postulate (v) by

(v′)
$$H_{N+M}(\mathcal{P}_{\mathbf{X}} \cup \mathcal{Q}_{\mathbf{Y}}) = \phi^{-1}\left[\frac{w(\mathcal{P}_{\mathbf{X}})\phi[H_N(\mathcal{P}_{\mathbf{X}})] + w(\mathcal{Q}_{\mathbf{Y}})\phi[H_M(\mathcal{Q}_{\mathbf{Y}})]}{w(\mathcal{P}_{\mathbf{X}}) + w(\mathcal{Q}_{\mathbf{Y}})}\right]. \tag{2.20}$$

The choice of the function ϕ is not arbitrary. In order to maintain additivity, the admissible choices for this function are: 1) $\phi(x) = x$, in which case (v′) in fact becomes (v), or 2) $\phi(x) = 2^{(1-\alpha)x}$. In the latter case, Renyi obtained the measure

$$H_{R_\alpha}(\mathbf{X}) = \frac{1}{1-\alpha}\log\sum_{i=1}^{N} p_i^{\alpha}, \tag{2.21}$$

for $\alpha > 0$ and $\alpha \neq 1$. Here, we have assumed complete probability distributions, that is $\sum_{i=1}^{N} p_i = 1$. Renyi called this the entropy of order α.

The Shannon entropy appears as a special case of the Renyi entropy, since [Renyi, 1976b,a]

$$\lim_{\alpha \to 1} H_{R_\alpha}(\mathbf{X}) = H(\mathbf{X}), \tag{2.22}$$

where $H(\mathbf{X})$ is the Shannon entropy given by Eq. (2.1).

The difference between the Renyi entropy and the Shannon entropy is that the Renyi entropy no longer satisfies the fourth basic postulate, that is, the recursive property. All the other properties are satisfied, including the additivity property, which Renyi considered to be the fundamental property of an information measure. See also [Aczél and Daróczy, 1975] for a detailed derivation of the Renyi entropy properties.

Of special interest to this dissertation is Renyi's entropy of order two, that is $\alpha = 2$, because this measure is quadratic in the distribution (the importance of this will become

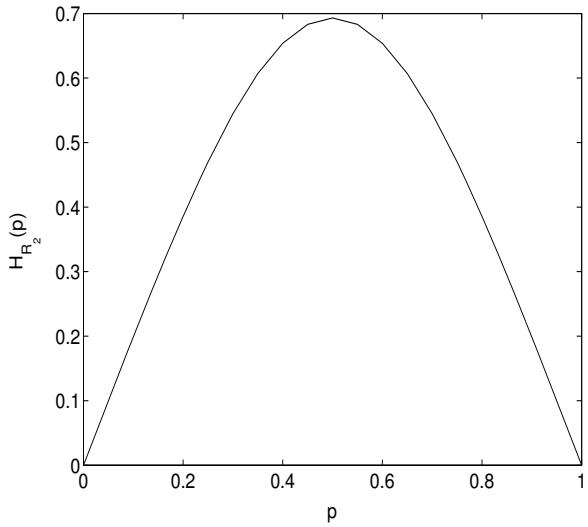


Figure 2.2: $H_{R_2}(p)$ *(nats) versus* p*.*

apparent later). We denote the resulting quantity by $H_{R_2}(\mathbf{X})$ and denote it by the *Renyi quadratic entropy*. It is given by

$$H_{R_2}(\mathbf{X}) = -\log \sum_{i=1}^{N} p_i^2. \tag{2.23}$$

Example

As an illustration of how the Renyi quadratic entropy relates to the Shannon entropy, we consider the simple example discussed in section 2.1. Assume again that a stochastic variable $\mathbf{X}$ can only take two outcomes, where $p_1 = p$ and $p_2 = 1 - p$. Hence,

$$H_{R_2}(\mathbf{X}) = -\log\left\{p^2 + (1-p)^2\right\} = H_{R_2}(p). \tag{2.24}$$

Figure 2.2 shows the graph of the function $H_{R_2}(p)$. We note that the graph has the same characteristic shape as the Shannon entropy had for this example.

Renyi Differential Entropy

The continuous counterpart of the Renyi entropy is the differential Renyi entropy. The order α differential Renyi entropy is given by

$$\begin{aligned} h_{R_\alpha}(\mathbf{X}) &= \frac{1}{1-\alpha}\log\int f^\alpha(\mathbf{x})d\mathbf{x}, \\ &= \frac{1}{1-\alpha}\log E_f\left\{f^{1-\alpha}(\mathbf{x})\right\}. \end{aligned} \tag{2.25}$$

Again, selecting $\alpha = 2$, we obtain the differential Renyi quadratic entropy

$$\begin{aligned} h_{R_2}(\mathbf{X}) &= -\log\int f^2(\mathbf{x})d\mathbf{x}, \\ &= -\log E_f\left\{f(\mathbf{x})\right\}. \end{aligned} \tag{2.26}$$

Some properties of the Renyi entropy of order α are the following [Hild, 2003]

1. Just as for the Shannon entropy, the Renyi entropy is maximum for a uniform distribution for random variables having finite support.

2. The Renyi entropy is not in general maximized by the Gaussian distribution in the fixed variance case.

3. The Renyi entropy is invariant to rotations and to translations.

It should be noted that other entropy measures besides the Shannon and Renyi definitions exist, such as Havrda-Charvat, Sharma and Mittal, Sharma-Taneja and Kapur's 1st and 2nd entropy definitions [Kapur and Kesavan, 1992, Kapur, 1994].

Renyi Divergence

Renyi also analyzed the properties of the Kullback-Leibler divergence. Just as in the Shannon entropy case, he expressed the Kullback-Leibler measure as a mean value, and identified some basic properties which should be satisfied for such a discrimination measure. Similarly as in the entropy case, he generalized the mean value property, and derived the following distance measure between the pdfs $f(\mathbf{x})$ and $g(\mathbf{x})$ [Renyi, 1976a]

$$\begin{aligned} D_{R_\alpha}\{f,g\} &= \frac{1}{1-\alpha}\log\int \frac{f^\alpha(\mathbf{x})}{g^{\alpha-1}(\mathbf{x})}d\mathbf{x}, \\ &= \frac{1}{1-\alpha}\log E_f\left\{\frac{f^{\alpha-1}(\mathbf{x})}{g^{\alpha-1}(\mathbf{x})}\right\}. \end{aligned} \tag{2.27}$$

The Renyi divergence possess the following properties [Renyi, 1976a]

1. $D_{R_\alpha}\{f,g\} \geq 0,\ \forall f,g,\alpha > 0.$

2. $D_{R_\alpha}\{f,g\} = 0,$ iff $f(\mathbf{x}) = g(\mathbf{x}),\ \forall \mathbf{x} \in R^d.$

3. $\lim_{\alpha\to 1} D_{R_\alpha}\{f,g\} = D_{KL}\{f,g\}.$

4. $D_{R_\alpha}\{f,g\}$ is additive for independent events.

It is easily seen that the Renyi divergence is not symmetric, and hence it is not a metric. It can also be seen that the Renyi divergence for $\alpha = 2$ is not quadratic in both the densities in the same way as the Renyi entropy for $\alpha = 2$. This implies for example that the Renyi quadratic divergence is not implicitly based on the Renyi quadratic entropy the same way that the Kullback-Leibler divergence is based on the Shannon entropy.

The Renyi divergence may also be used as a measure of mutual information between random variables, by considering the divergence between the joint density and the product of marginal densities. Note that the Renyi mutual information can not be expressed in terms of Renyi entropies in the same manner as the Shannon mutual information can in terms of Shannon entropies (see Eq. (2.18)).

Other Divergence Measures

It should be noted that there are many other important distance measures between pdfs apart from the Kullback-Leibler divergence and the Renyi divergence. Common for all these measures is that the term "distance" is used loosely, since the symmetry and triangular properties may not always be satisfied. These measures may not even be "information measures" in the sense that properties like additivity may not be satisfied. Even so, we will refer to such measures as information theoretic quantities, since they all obviously convey some sort of information about the closeness of two probability distributions. Some such measures are [Kazakos and Papantoni-Kazakos, 1990]

$$D_J(f,g) = \frac{D_{KL}\{f,g\} + D_{KL}\{g,f\}}{2}, \tag{2.28}$$

which is called the Jeffrey's distance [Jeffreys, 1948]. Jeffrey's distance is a symmetric version of the Kullback-Leibler measure. Another measure is due to Chernoff [1952], defined as

$$D_C(f,g) = -\log \int f^{1-t}(\mathbf{x}) q^t(\mathbf{x}) d\mathbf{x}, \quad 0 \leq t \leq 1, \tag{2.29}$$

where the most well known member is the Bhattacharyya [1943] distance for $t = \frac{1}{2}$. Also, a measure known as the variational distance may be used. This is defined as

$$D_V(f,g) = \int |f(\mathbf{x}) - g(\mathbf{x})| \, d\mathbf{x}. \tag{2.30}$$

We will not consider any of these divergence measures in more detail. Which measure to use will not only depend on their mathematical properties, but also depend upon other requirements such as ease of implementation.

2.3 Quadratic Divergence Measures

In this section we review some recently introduced divergence measures between pdfs proposed by Principe et al. [2000a].

Cauchy-Schwarz Divergence

Principe et al. [2000a] defined a pdf divergence measure based on the Cauchy-Schwarz (CS) inequality. Define the inner-product between two square-integrable functions $f(\mathbf{x})$ and $g(\mathbf{x})$ as $\langle f, g \rangle = \int f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x}$. Then, by the Cauchy-Schwarz inequality

$$\left| \int f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x} \right|^2 \leq \int |f(\mathbf{x})|^2 \, d\mathbf{x} \int |g(\mathbf{x})|^2 \, d\mathbf{x}, \tag{2.31}$$

with equality if and only if the two functions are linearly dependent. Let $f(\mathbf{x})$ and $g(\mathbf{x})$ be pdfs, i.e. non-negative and integrating to unity. Then, the Cauchy-Schwarz pdf divergence is defined as [Principe et al., 2000a]

$$\begin{aligned} D_{CS}\{f,g\} &= -\log \frac{\int f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x}}{\sqrt{\int f^2(\mathbf{x}) d\mathbf{x} \int g^2(\mathbf{x}) d\mathbf{x}}} \\ &= -\log \frac{E_f\{g(\mathbf{x})\}}{\sqrt{E_f\{f(\mathbf{x})\} E_g\{g(\mathbf{x})\}}}. \end{aligned} \tag{2.32}$$

The Cauchy-Schwarz pdf distance has the following properties

1. $D_{CS}\{f,g\} \geq 0,\ \forall f,g,\alpha > 0$.
2. $D_{CS}\{f,g\} = 0,$ iff $f(\mathbf{x}) = g(\mathbf{x}),\ \forall \mathbf{x} \in R^d$.
3. $D_{CS}\{f,g\} = D_{CS}\{g,f\}$.
4. $D_{CS}\{f,g\}$ is additive for independent events.

The first three properties were proved by [Xu, 1999]. The last property has not been proved earlier, so we will prove it as follows. For simplicity, we consider the case where $\mathbf{X} = [X_1, X_2]^T$, such that $f(\mathbf{x}) = f(x_1)f(x_2)$, $g(\mathbf{x}) = g(x_1)g(x_2)$ by independence.

Proof of property 4.

$$\begin{aligned}
D_{CS}\{f(\mathbf{x}), g(\mathbf{x})\} &= -\log \frac{\int\int f(x_1)f(x_2)g(x_1)g(x_2)dx_1dx_2}{\sqrt{\int\int [f(x_1)f(x_2)]^2dx_1dx_2 \int\int [g(x_1)g(x_2)]^2dx_1dx_2}} \\
&= -\log \frac{\int f(x_1)g(x_1)dx_1 \int f(x_2)g(x_2)dx_2}{\sqrt{\int f^2(x_1)dx_1 \int g^2(x_1)dx_1 \int f^2(x_2)dx_2 \int g^2(x_2)dx_2}} \\
&= -\log \frac{\int f(x_1)g(x_1)dx_1}{\sqrt{\int f^2(x_1)dx_1 \int g^2(x_1)dx_1}} - \log \frac{\int f(x_2)g(x_2)dx_2}{\sqrt{\int f^2(x_2)dx_2 \int g^2(x_2)dx_2}} \\
&= D_{CS}\{f(x_1), g(x_1)\} + D_{CS}\{f(x_2), g(x_2)\}, \qquad (2.33)
\end{aligned}$$

which completes the proof.

Still, the CS divergence does not satisfy the triangle inequality, and for this reason it is not a distance metric.

The CS divergence can be used as a measure of statistical independence between random variables, as shown for example in [Principe et al., 2000a, Xu, 1999].

It can be seen that the CS divergence is implicitly based on Renyi's quadratic entropy, since

$$D_{CS}\{f,g\} = -\log \int f(\mathbf{x})g(\mathbf{x})d\mathbf{x} - \frac{1}{2}\left(-\log \int f^2(\mathbf{x})d\mathbf{x}\right) - \frac{1}{2}\left(-\log \int g^2(\mathbf{x})d\mathbf{x}\right), \quad (2.34)$$

where $-\log \int f^2(\mathbf{x})d\mathbf{x}$ is the Renyi quadratic entropy with respect to $f(\mathbf{x})$, and $-\log \int g^2(\mathbf{x})d\mathbf{x}$ is the Renyi quadratic entropy with respect to $g(\mathbf{x})$. The term $-\log \int f(\mathbf{x})g(\mathbf{x})d\mathbf{x}$ can be interpreted as the "cross-entropy" between $f(\mathbf{x})$ and $g(\mathbf{x})$.

Integrated Squared Error

Principe et al. [2000a] also proposed an integrated squared error (ISE) divergence measure between the two pdfs $f(\mathbf{x})$ and $q(\mathbf{x})$, defined as

$$\begin{aligned}
D_{\text{ISE}}\{f,g\} &= \int [f(\mathbf{x}) - g(\mathbf{x})]^2 d\mathbf{x}, \\
&= \int f^2(\mathbf{x})d\mathbf{x} - 2\int f(\mathbf{x})g(\mathbf{x})d\mathbf{x} + \int g^2(\mathbf{x})d\mathbf{x}, \\
&= E_f\{f(\mathbf{x})\} - 2E_f\{g(\mathbf{x})\} + E_g\{g(\mathbf{x})\}. \qquad (2.35)
\end{aligned}$$

This measure also vanishes only if the two pdfs in question are identical, it is always non-negative and also symmetric. It does not satisfy the additivity property, and for that reason one should be careful when dubbing this measure *information theoretic*. It can be seen that the terms involved in Eq. (2.35) are intrinsically related to Renyi's quadratic entropy.

2.4 Estimating Information Theoretic Quantities

Since the analytical expressions of the data distributions are normally not known, one needs to estimate the information theoretic measures based on the available data samples. Sample estimators for information theoretic quantities typically rely on the *plug-in* a density estimator principle. That is, using the available samples, one needs to obtain an estimate of the underlying probability density distributions, which in turn is substituted into the cost function expression.

There are two possible techniques one can assume toward estimating the pdf of a random variable from its independent and identically distributed (iid) samples: Parametric and non-parametric. In the following, we briefly discuss some of the most well-known methods in these categories. We illustrate the different methods in the univariate case. For some of these methods, the extension to the multivariate case is straightforward, for others it is more difficult. However, in applications like ICA, we rarely need to estimate densities in more than one dimension. Of special importance to this dissertation is a method known as Parzen windowing. This method will be discussed also in the multivariate case.

Parametric Density Estimation

Assume that the densities are given in a parametric form, and that the corresponding parameters form the vector θ which is unknown. Let $x_1, \ldots, x_N$ be random samples drawn from pdf $f(x;\theta)$, where the dependence on θ has been explicitly shown.

Maximum Likelihood Estimator

The joint pdf $f(X;\theta)$, is formed, where $\mathcal{X} = \{x_1, \ldots, x_N\}$ is the set of samples. By statistical independence, $f(\mathcal{X};\theta) = \prod_{i=1}^{N} f(x_i;\theta)$, which is known as the likelihood function of θ. The maximum likelihood (ML) method estimates θ such that the likelihood function takes its maximum value, that is [Theodoridis and Koutroumbas, 1999]

$$\hat{\theta}_{\text{ML}} = arg\max_{\theta} \prod_{i=1}^{N} f(x_i;\theta). \tag{2.36}$$

In practice, the log-likelihood is used, since it is easier to compute the gradient of this function. In a similar manner, the maximum a posteriori probability (MAP) estimate $\hat{\theta}_{\text{MAP}}$ is defined as the point where $f(\theta|\mathcal{X}) = f(\theta)f(\mathcal{X}|\theta)$ becomes maximum.

Mixture Models

More flexibility is incorporated into the density estimation by the use of mixture models. Here, the unknown density is modeled as a mixture of J densities

$$f(x) = \sum_{j=1}^{J} f(x|j)P_j, \tag{2.37}$$

where $\sum_{j=1}^{J} P_j = 1$. Thus, this modeling assumes that each point x may be "drawn" from any of the J model distributions with probability P_j, $j = 1, \ldots, J$. Now, the density components are given a parametric form, $f(x|j;\theta)$, and then the unknown parameters θ and P_j, $j = 1, \ldots, J$ must be computed from the samples. Since the contribution of mixture density is not known, the maximum likelihood principle can not be easily employed, and one needs to resort to the expectation-maximization (EM) algorithm [McLachlan and Peel, 2000] to solve the problem. Because of the additional flexibility the mixture models add to the parametric models, this method may be regarded as semi-parametric.

There are basically two problems associated with the parametric density estimation schemes discussed above. In the case that an information theoretic measure is used as a cost function for training adaptive systems, the parametric methods requires solving an optimization problem within an optimization problem, the latter being the adaption process (using for example gradient-based learning algorithms). The second drawback of the parametric approach is the insufficiency of parametric models for general-purpose modeling tasks. The selected parametric family may be to simplistic to be able to accurately model the data distributions in question, and it may be difficult to select the best parametric class for the problem at hand.

Non-Parametric Density Estimation

To avoid having to select a parametric model for the densities, one often resort non-parametric density estimation methods.

Histograms

The oldest and most widely used density estimator is the histogram [Silverman, 1986]. Given an origin x_0 and a bin width h, the bins of the histogram are defined as $[x_0+mh, x_0+(m+1)h)$ for positive and negative integers m. The histogram is then defined as [Silverman, 1986]

$$\hat{f}(x) = \frac{1}{Nh}(\text{no. of } x_i \text{ in same bin as } x) \tag{2.38}$$

The discontinuity of histograms causes difficulty if derivatives of the estimates are required. Also, the choice of origin can affect the histogram estimate significantly.

Naive Density Estimator

Another variant of the histogram can be derived as follows. A probability density function evaluated at x can be defined as

$$f(x) = \lim_{h \to 0} \frac{1}{2h} P(x - h < X < x + h). \tag{2.39}$$

The probability $P(x-h<X<x+h)$ can be estimated by counting the number of data samples falling into a bin of size $2h$ centered at x. This can be expressed more precisely by defining a weight function I as

$$I=\begin{cases}\frac{1}{2}, & \text{if } |x|<1\\ 0, & \text{otherwise}\end{cases} \tag{2.40}$$

such that the *naive estimator* [Silverman, 1986] can be expressed as

$$\hat{f}(x)=\frac{1}{Nh}\sum_{i=1}^{N} I\left(\frac{x-x_i}{h}\right). \tag{2.41}$$

Hence, this estimator is constructed by placing a "box" of width $2h$ and height of $(2Nh)^{-1}$ on each observation and then summing to obtain the estimate.

This variant of the histogram does not suffer from the difficulty of having to choose an origin. The choice of bin width still remains and is governed by the parameter h. As can be seen, the resulting estimate is not a continuous function.

Parzen Window Density Estimator

The naive estimator can be generalized by replacing the weight function by a non-negative kernel function K satisfying the condition,

$$\int K(x)dx=1. \tag{2.42}$$

Often K is chosen to be a symmetric, unimodal pdf such as the standard normal density. The resulting estimator is called the kernel density estimator, or the Parzen window estimator. It is given by [Parzen, 1962]

$$\hat{f}(x)=\frac{1}{N\sigma}\sum_{i=1}^{N} K\left(\frac{x-x_i}{\sigma}\right)=\frac{1}{N}\sum_{i=1}^{N} W_\sigma(x-x_i), \tag{2.43}$$

where

$$W_\sigma(x-x_i)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x-x_i)^2}{2\sigma^2}\right\}, \tag{2.44}$$

a Gaussian density situated at x_i having variance, or window width, σ^2 (in the case that K is a standard normal density). This is the so-called Parzen window. Using such a kernel function, it follows from the definition that $\hat{f}(x)$ will itself be a probability density. Moreover, $\hat{f}(x)$ will inherit all the continuity and differentiability properties of $W_\sigma(x)$, such that it will be a smooth curve having derivatives of all orders [Silverman, 1986]. The Parzen window need not be Gaussian. Other kernel choices also exist, such as the triangle, Epanechnikov, biweight or triweight kernels [Scott, 1992].

It is easily shown that the Parzen window density estimator is asymptotically unbiased and consistent provided the kernel size $\sigma(N)$ is annealed toward zero at a sufficiently low rate as N tends to infinity [Parzen, 1962, Silverman, 1986, Scott, 1992, Wand and Jones, 1995]. In the finite sample case, the kernel size has to be chosen in a trade-off between estimation bias and variance, as we discuss next.

It is common to measure the closeness of the estimator $\hat{f}(x)$ to the target density $f(x)$ in the point x by the size of the mean squared error (MSE)

$$MSE\left\{\hat{f}(x)\right\} = E\left\{[\hat{f}(x) - f(x)]^2\right\}. \tag{2.45}$$

The MSE decomposes into a bias term and a variance term

$$MSE\left\{\hat{f}(x)\right\} = \left[E\left\{\hat{f}(x)\right\} - f(x)\right]^2 + Var\left\{\hat{f}(x)\right\}. \tag{2.46}$$

Rather than just estimating $f(x)$ at a fixed point, it is more desirable to estimate $f(x)$ over the whole x-space. The mean integrated squared error (MISE) is thus the appropriate measure for analyzing $\hat{f}(x)$, where

$$MISE\left\{\hat{f}(x)\right\} = \int \left[E\left\{\hat{f}(x)\right\} - f(x)\right]^2 dx + \int Var\left\{\hat{f}(x)\right\} dx. \tag{2.47}$$

The bias term and the variance term in Eq. (2.47) depends on the kernel size σ in very different ways. This is best illustrated by analyzing Eq. (2.47) asymptotically, that is, for large sample sizes N. Under mild conditions on $f(x)$, using a smooth kernel function K such as the standard Gaussian density, and under the condition that $\sigma(N)$ approaches zero at a rate slower than N^{-1}, it can be shown that the asymptotic mean integrated squared error (AMISE) is given by [Wand and Jones, 1995]

$$AMISE\left\{\hat{f}(x)\right\} = \frac{\sigma^4 \mu_2^2(K) R(f'')}{4} + \frac{R(K)}{\sigma N}, \tag{2.48}$$

where $\mu_2(K) = \int z^2 K(z) dz$, $R(f'') = \int \{f''(x)\}^2 dx$ where $f''(x) = \frac{d^2}{dx^2} f(x)$, and $R(K) = \int K(z)^2 d\mathbf{z}$. It can be seen that the left term on the right-hand side of Eq. (2.48) is minimized by minimizing σ. This is the bias part. However, the right term, which is the variance part, is minimized by maximizing σ. Hence, there is an inherent bias-variance trade-off associated with the Parzen window technique for density estimation. There exist several methods for selecting the kernel size σ, each having its own properties [Wand and Jones, 1995]. Here, we mention two of the most well-known and easiest to use.

The first method we examine utilizes Eq. (2.48) for kernel size selection. Note that one may obtain an explicit formula for the AMISE optimal kernel size by differentiating Eq. (2.48) and equating it to zero, obtaining

$$\sigma_{\text{AMISE}} = \left[\frac{R(K)}{\mu_2^2(K) R(f'') N}\right]^{\frac{1}{5}}. \tag{2.49}$$

One straight-forward approach is to estimate $R(f'')$ by assuming that the true underlying density is a normal density. This quantity is then plugged back into Eq. (2.48) to obtain an estimate for σ_{AMISE}. It can be shown that the corresponding kernel size is given by [Silverman, 1986]

$$\hat{\sigma}_{\text{AMISE}} \approx 1.06 \hat{\sigma} N^{-\frac{1}{5}}, \tag{2.50}$$

where $\hat{\sigma}$ is an estimate of the standard deviation of the normal density. The main appeal of the normal reference rule is that is is very easy to use. The obvious drawback is that it assumes that the underlying density is unimodal and Gaussian. Therefore, using the normal

reference rule will often result in selecting a kernel size which is too large, thus corresponding to a oversmoothed density estimate.

The earliest fully automatic kernel size selector was based on the idea of cross-validation. The aim is to find the kernel size that minimizes the MISE. Minimizing the MISE is equivalent to minimizing

$$MISE\left\{\hat{f}(x)\right\} - \int f^2(x)dx = E\left[\int \hat{f}^2(x) - 2\int \hat{f}(x)f(x)dx\right]. \tag{2.51}$$

The right-hand side is unknown sine it depends on $f(x)$. However, an unbiased estimator for this quantity is [Wand and Jones, 1995]

$$LSCV(\sigma) = \int \hat{f}^2(x) - 2N^{-1}\sum_{i=1}^{N}\hat{f}_{-i}(x_i), \tag{2.52}$$

where $\hat{f}_{-i}(x) = \frac{1}{N-1}\sum_{j\neq i} W_\sigma(x - x_j)$ is the density estimate based on the sample x_i deleted. Now, one searches for the kernel size which minimizes $LSCV(\sigma)$. The search can be performed over a range of σ-values. The upper limit for the search may be defined as the value obtained by using the normal reference rule, since this value is likely too large to begin with. Minimizing $LSCV(\sigma)$ is called least squares cross-validation (LSCV), which refers to the use of a part of a sample to obtain information about another part. The resulting kernel size σ_{LSCV} has been analyzed and found to have a large variance [Scott, 1992]. This is a result of the bias-variance trade-off, keeping in mind that it is an unbiased estimator. Thus, this estimator has often been found to produce a too small kernel size [Wand and Jones, 1995].

More advanced kernel size selectors are discussed for example in [Sheater and Jones, 1991, Jones et al., 1996, Wand and Jones, 1995, Scott, 1992].

The Parzen window density estimator is easily extended to the multivariate case. The d-dimensional kernel function $K(\mathbf{x})$ must integrate to one. Often it is chosen to be the standard multivariate normal density function, which gives

$$\hat{f}(\mathbf{x}) = \frac{1}{N\sigma^d}\sum_{i=1}^{N} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{\sigma}\right) = \frac{1}{N}\sum_{i=1}^{N} W_\sigma(\mathbf{x}-\mathbf{x}_i), \tag{2.53}$$

where

$$W_\sigma(\mathbf{x}-\mathbf{x}_i) = \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}}\exp\left\{-\frac{||\mathbf{x}-\mathbf{x}_i||^2}{2\sigma^2}\right\}. \tag{2.54}$$

Again, other kernel functions besides the Gaussian may be used. Also, using a spherical Gaussian kernel function, like $W_\sigma(\mathbf{x}-\mathbf{x}_i)$ as defined above, implies that the same window width is appropriate for all the data dimensions. This may not be the case in general, which requires the specification of many more bandwidth parameters than in the univariate setting. One may for example use a Gaussian kernel function specified by a covariance matrix $\mathbf{\Sigma}$. Naturally, this makes the process of determining an appropriate $\mathbf{\Sigma}$ more complicated than determining a scalar σ. We will not consider this more complicated problem further.

The least squares cross-validation technique and the normal reference rule apply directly also in the multivariate case. The normal reference rule becomes [Silverman, 1986]

$$\sigma_{\text{AMISE}} = \hat{\sigma}\left[\frac{4}{(2d+1)N}\right]^{\frac{1}{d+4}}, \tag{2.55}$$

where $\hat{\sigma}^2 = d^{-1} \sum_i \mathbf{\Sigma}_{ii}$, and $\mathbf{\Sigma}_{ii}$ are the diagonal elements of the sample covariance matrix.

However, one major problem with multivariate Parzen window density estimation is due to the "curse-of-dimensionality". The curse-of-dimensionality refers to the fact that the usual bias-variance trade-off cannot be accomplished very well in higher dimensions without very large samples.

Orthogonal Series Density Estimators

This method assumes that a square-integrable $f(x)$ can be represented as a convergent orthogonal series expansion [Izenman, 1991, Devroye, 1989]

$$f(x) = \sum_{k=1}^{\infty} a_k \phi_k(x), \quad x \in \Omega, \tag{2.56}$$

where $\{\phi_k\}$ is a complete orthonormal system of functions on a set Ω on the real line, that is $\int_\Omega \phi_j(x)\phi_k(x)dx = \delta_{jk}$, where δ_{jk} is the Kronecker delta. The coefficients $\{a_k\}$ are defined by $a_k = E_f[\phi_k(X)]$. Orthonormal systems proposed for $\{\phi_k\}$ are those with compact support, such as Fourier, trigonometric and Haar systems on $[0, 1]$, and the Legendre system on $[-1, 1]$. Those with unbounded support are the Hermite system (Gram-Charlier, Edgeworth) on R and the Laguerre system on $[0, \infty)$.

The Gram-Charlier and Edgeworth expansions are well-known for example in ICA [Comon, 1994, Amari et al., 1996]. These differ in the ordering of the terms in the expansion. The Gram-Charlier series uses [Izenman, 1991, Haykin, 1999]

$$\begin{aligned} \phi_k(x) &= \frac{1}{(2^k k! \pi^{\frac{1}{2}})^{\frac{1}{2}}} \exp\left\{-\frac{x^2}{2}\right\} H_k(x), \\ H_k(x) &= = (-1)^k \exp\left\{-\frac{x^2}{2}\right\} \frac{d^k}{dx^k} \exp\left\{-x^2\right\}, \end{aligned} \tag{2.57}$$

where $H_k(x)$ is the kth Hermite polynomial. Hence, the Gram-Charlier expansion is expressed in terms of a Gaussian reference density and its derivatives. Because of the use of a reference density, one can argue that this density estimation methods perhaps should be called semi-parametric.

In practice, the orthogonal series density estimators are truncated, so that only a few terms are retained. This may lead to pdf estimates that do not satisfy the two basic conditions for a function to be a probability distribution: non-negativity and integration to one.

2.5 Information Theoretic Clustering

Clustering refers to the problem of partitioning a data set into subsets, such that members within subsets are more "similar" to each other, according to some criterion, than to members of other subsets. Each subset is called a cluster. The clustering should be accomplished without any prior knowledge about the structure of the data, and in general even without knowing the number of clusters which comprise the data.

Clustering is recognized as an important tool in areas such as image segmentation [Mohanta et al., 2002, Guo and Ma, 1998b,a], speech recognition [Watanabe et al., 2004, Goddard et al., 2000, Faltlhauser and Ruske, 2001], signal compression [Arrowood and Clements, 2004,

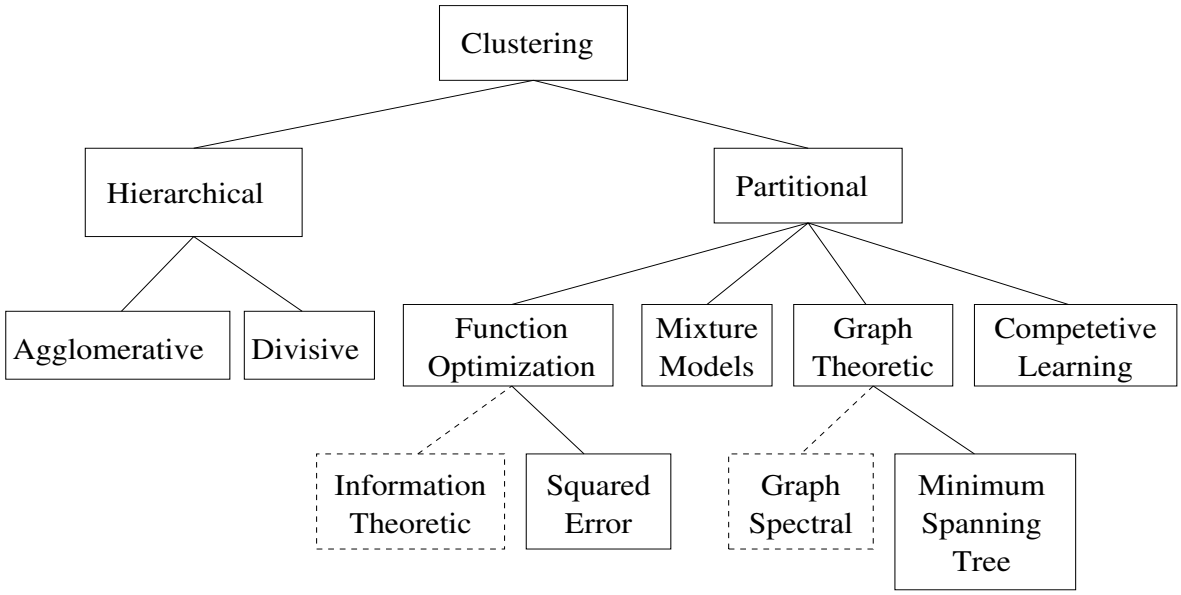


Figure 2.3: *Taxonomy of clustering approaches.*

Siala and Benazza-Benyahia, 2004, Mielikainen and Toivanen, 2003], medical research [Rigau et al., 2004, Greene et al., 2004, Shen et al., 2003], document storage and retrieval [Kogan et al., 2003, Dhillon and Modha, 2001, Dhillon et al., 2003a], world-wide-web search and mining [Huang et al., 2004, He et al., 2002, Zeng et al., 2002] and bioinformatics [Zhou et al., 2004, Li et al., 2004, Herrero et al., 2001], to name a few.

The clustering problem has a long history, not only in pattern recognition, but also in fields such as biology, psychiatry, psychology, archeology and geology [Jain and Dubes, 1988]. Yet, the problem is in itself ill-posed, since the goal is to find a "reasonable" partition of the data points. What is meant by "reasonable" depends on the application, the representation of the data and the assumptions about the origins of the data points [Tishby and Slonim, 2001]. Therefore, there is no clustering technique that is universally applicable in uncovering the structures present in multi-dimensional data sets, resulting in a clustering literature which is huge and varied. There are a number of books available, some dating back to the 1970's [Anderberg, 1973, Hartigan, 1975]. More recent books include [Jain and Dubes, 1988, Theodoridis and Koutroumbas, 1999, Duda et al., 2001].

It is impossible to cover all kinds of clustering algorithms. A taxonomy anno 1999 of the most well-known clustering methods was given by Jain et al. [1999], and to a large extent paralleled by Theodoridis and Koutroumbas [1999] in their treatment of clustering techniques. Fig. 2.3 shows this taxonomy. The stapled boxes were not in the Jain et al. [1999] version. These will also be discussed in this section.

The hierarchical clustering algorithms produce N levels of groupings, where N is the number of data patterns. These methods yield a *dendogram* [Jain and Dubes, 1988], representing the similarity levels at which groupings change. The dendogram may be used to determine the number of clusters in the data set. The agglomerative approach begins with each pattern in a distinct cluster, and successively merges clusters together. The divisive method begins with all patterns in a single cluster and performs a splitting operation. Most hierarchical clustering algorithms are variants of the single-link [Sneath and Sokal, 1973],

complete-link [King, 1967] and minimum variance [Ward, 1963, Murtagh, 1984] algorithms. These algorithms differ in the way they characterize the similarity between a pair of clusters. For example, the single-link algorithm may handle well-separated, non-isotropic, chain-like clusters very well. The complete-link algorithm is best suited data sets having isotropic, Gaussian-like, clusters.

The partitional clustering algorithms obtains a single partition of the data instead of a cluster hierarchy. They often need the number of clusters to be specified *a-priori*.

Competitive learning has been found to exist in biological neural networks. Artificial neural networks have therefore been used to group similar data patterns using competitive learning, and to represent these groups by a single neuron. The network weights are iteratively changed based on data correlations until a termination criterion is satisfied. Examples of such single-layered networks are the Kohonen [1989] learning vector quantization and self-organizing map, and adaptive resonance theory models [Carpenter and Grossberg, 1990]. These networks are only suitable for detecting hyperspherical clusters, and depend heavily on user-specified parameters. A two-layered cluster-detection-and-labeling network was proposed by Eltoft and deFigueiredo [1998], which is capable of clustering data sets irrespective of the cluster shapes.

The assumption underlying the mixture models is that the patterns to be clustered are drawn from one of several parametric distributions [McLachlan and Basford, 1988, McLachlan and Peel, 2000]. Hence, the overall probability distribution is a mixture of these component distributions, which are most often chosen to be Gaussian [Banfield and Raftery, 1993]. The fit between the data and the model is then optimized using the EM algorithm [Dempster et al., 1977].

Traditionally, the most well-known graph theoretic partitional clustering algorithm is based on construction of the minimal spanning tree (MST) of the data [Zahn, 1971]. The clusters are generated by deleting the MST edges with the largest lengths. For an appropriately constructed MST, this algorithm may handle a wide variety of data structures.

During the last 5-10 years a new line of research in graph-based clustering has emerged, which is considered the state-of-the art by many researchers. It is based on the *spectral* properties of graphs. The interest in these methods are so recent that they weren't a part of the Jain et al. [1999] taxonomy. Even so, graph spectral clustering dates back at least to Fiedler [1973], who discovered that a graph can be bi-partitioned by thresholding the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix. During the last few years, this method has been rediscovered, and a number of related techniques have been published [Ding et al., 2001, Shi and Malik, 2000, Perona and Freeman, 1998, Hagen and Kahng, 1991, Pothen et al., 1990, Sarkar and Soundararajan, 2000]. These techniques are all based on variants of the graph *cut*, a measure of the cost of partitioning a graph into two pieces. But they use different graph matrices, and utilize the information contained in the eigenspectrum of the matrices in different manners. Multiway cuts have also been studied [Chang et al., 1994, Meila and Xu, 2004]. For other related work, see for example [Weiss, 1999, Kannan et al., 2000, Alpert and Yao, 1995, Azar et al., 2001, Scott and Longuet-Higgins, 1990, Higham and Kibble, 2004, Jenssen et al., 2004b]. Another direction in graph spectral clustering was proposed by Ng et al. [2002]. In that work they use the eigenvectors of the Laplacian matrix to transform, or map, the input data into a new representation, for then to perform the actual clustering in that space by the K-means technique [MacQueen, 1967], which we discuss next.

Since the 1960's, the most intuitive and most frequently used *criterion function* in parti-

tional clustering techniques is the squared error criterion, defined as

$$J(\mathbf{U}, \mathbf{C}) = \sum_{i=1}^{N} \sum_{j=1}^{K} u_{ij} ||\mathbf{x}_i - \mathbf{c}_j||^2. \tag{2.58}$$

Here, $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_K]$, where $\mathbf{c}_j,\ j = 1, \ldots, K$, is the centroid of cluster C_j. $\mathbf{U}$ is a $(N \times K)$ matrix such that the (i, j) element $u_{ij} = 1$ if $\mathbf{x}_i$ is associated with cluster C_j, and zero otherwise. The goal is to adjust $\mathbf{U}$ and $\mathbf{C}$ such that $J(\mathbf{U}, \mathbf{C})$ is minimized.

The K-means algorithm [MacQueen, 1967] is the simplest and most commonly used algorithm for squared error clustering. It starts with a random initial partitioning and keeps reassigning the patterns to clusters based on the similarity between the patterns and the cluster centroids until a convergence criterion is met. It can be summarized as [Jain et al., 2000]

1. Select an initial partition with K clusters.
2. Generate a new partition by assigning each pattern to its closest cluster center.
3. Compute new cluster centers as the centroids of the clusters.
4. Repeat step 2 and 3 until convergence.

One great advantage of the K-means algorithm is its computational simplicity which scales linearly with the number of data patterns. It is however very sensitive to the initialization. The K-means algorithm and the related ISODATA algorithm [Ball and Hall, 1965] have given rise to several extended versions [Anderberg, 1973, Linde et al., 1980, Lloyd, 1982, Selim and Ismail, 1984, Diday, 1973, Symon, 1977], many of which try to resolve the initialization problem.

A very influential development was the derivation of fuzzy K-means [Bezdek, 1980, Bezdek et al., 1999, Cannon et al., 1986, Hathaway and Bezdek, 1986, Hathaway et al., 1989, Ismail and Selim, 1986]. In fuzzy K-means, the membership values are allowed to vary between zero and one, such that $u_{ij} \in [0, 1]$, with the requirement that $\sum_{j=1}^{K} u_{ij} = 1$, $i = 1, \ldots, N$. Hence, a data pattern can be associated with several clusters at the same time. Fuzzyfying the u_{ij}'s makes $J(\mathbf{U}, \mathbf{C})$ differentiable with respect to both $\mathbf{U}$ and $\mathbf{C}$, and optimization techniques developed in differential calculus may be used.

The K-means algorithm, and its variants, only work well for well-separated clusters which are hyperspherical in shape (hyperelliptical if the Mahalanobis distance is used instead of the squared error [Mao and Jain, 1996]). The reason is that the cost function is based on a minimum variance criterion with respect to the cluster centroids. For that reason, K-means has been referred to as the minimum variance partition [Jain et al., 2000], only capturing the second order statistics in the data.

From a statistical pattern recognition viewpoint, discovering clusters having non-Gaussian shapes requires a clustering cost function which captures all the statistical information contained in the data. *Information theoretic criteria* do capture all the data statistics, since they are functions of pdfs. Several different attempts have been made over the last 10-15 years to perform clustering based on information theoretic quantities. Such methods are emerging but not as rapidly as graph spectral clustering. Information theoretic clustering methods were not treated in the Jain et al. [1999] survey.

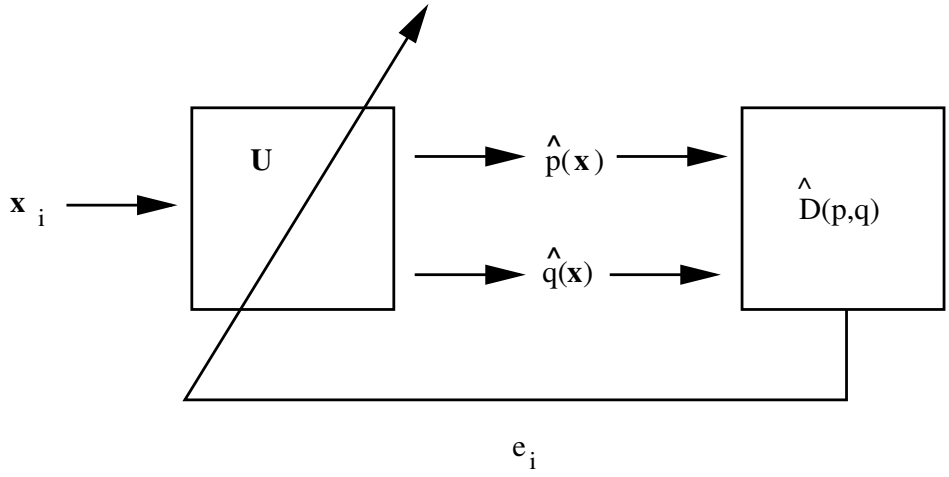


Figure 2.4: *Information theoretic clustering block-diagram.*

Rose et al. [1990, 1992, 1993, 1994] suggested a deterministic annealing approach, where sample points were associated with cluster representatives according to a maximum entropy probability distribution. A related approach for pairwise clustering was proposed by Hofmann and Buhmann [1997] based on mean-field approximation as minimization of Kullback-Leibler divergence. Similar in spirit was the algorithm proposed by Shimoji and Lee [1994]. Roberts et al. [1999, 2000, 2001] proposed to minimize the expected entropy of the partitions over the observed data, and modeled the cluster pdfs using a Gaussian mixture model. Recently, Dhillon et al. [2003b,c, 2002] proposed a global clustering cost function based on the Jensen-Shannon divergence, and derived a hierarchical clustering algorithm for optimizing it. See also [Banerjee et al., 2004]. Also, an approach based on mutual information has received a lot of attention lately. It is called the *information bottleneck* method [Tishby et al., 1999], and is derived as a generalization to rate distortion theory. It has been widely applied [Tishby and Slonim, 2001, Slonim and Tishby, 2000, Still et al., 2004, Gondek and Hofmann, 2003, Friedman et al., 2001, El-Yaniv and Souroujon, 2001]. See also [Fred and Jain, 2003, Kraskov et al., 2004] for recent mutual information-based clustering approaches. The impressive results obtained by the aforementioned clustering techniques clearly has shown that clustering based on information theoretic criteria has great potential.

Gokcay and Principe [2000] proposed a different approach to information theoretic clustering. They proposed to assign data patterns to the clusters in such a way that a distance measure between the estimated pdfs is as large as possible. A block-diagram of such a procedure in the two-cluster case is shown in Fig. 2.4. In the multi-cluster case, the divergence measure must be extended, or a recursive clustering approach must be applied. Note that the final goal of this procedure is to obtain a "reasonable" partition of the data set according to the chosen pdf divergence measure, and *not* to necessarily estimate the cluster pdfs *perfectly*, even though density estimation is an intermediate step. That is, the cluster pdfs need to be estimated "good enough" relative to each other, to obtain a reasonable partition. Such an approach is somewhat similar in spirit to density-based algorithms like [Fukunaga, 1990, Ester et al., 1996, Ankerst et al., 1999, Hinneburg and Keim, 1998].

In theory, this approach makes no assumptions with respect to the choice of pdf distance measure or to the choice of density estimation procedure. However, using parametric den-

sity estimation techniques is not trivial when it comes to evaluating information theoretic cost functions. The reason is that they are expressed as integrals over functions of pdfs, which requires numerical integration procedures to be developed. Hence, it may be necessary to impose too strict assumptions about the cluster pdfs in order to evaluate the pdf distance measures.

Gokcay and Principe [2000] proposed to use the Cauchy-Schwarz pdf divergence measure integrated with Parzen windowing, because it may thus be easily estimated nonparametrically. However, the algorithm developed in [Gokcay and Principe, 2002] was not practical. It was deterministic in nature, and required all clustering possibilities to be examined. Hence, the computational load was exhaustive. The results obtained on small manageable datasets were however very promising, and demonstrated the potential of non-parametric, information theoretic criteria for clustering. Indeed, a major part of this thesis consists of developing practical information theoretic clustering algorithms using the Cauchy-Schwarz divergence, and to connect the CS divergence measure to graph spectral clustering and to Mercer kernel-based clustering. These topics will be treated in paper 1 and paper 2.

2.6 Independent Component Analysis

Independent component analysis has been developed over the last 10-15 years as a generative model for observed multivariate data, which are assumed to be mixtures of some unknown latent variables. Originally, it emerged as a means to solve the problem of blind source separation (BSS) [Jutten, 2000, Comon et al., 1996, Sorouchyari, 1991]. BSS refers to the problem of finding the original source signals from available mixtures, without any prior knowledge of the number of sources or the mixing mechanisms. In the linear case, the N observed signals at time instant i can be collected in the $(N \times 1)$ vector $\mathbf{x}_i$, which can be expressed as

$$\mathbf{x}_i = \mathbf{A}\mathbf{s}_i, \tag{2.59}$$

where the $(N \times N)$ matrix $\mathbf{A}$ is called the mixing matrix and the $(N \times 1)$ vector $\mathbf{s}_i$ contains the N source signals at time i (we have for simplicity assumed the same number of sources as sensors). Based on the assumption that the source signals are *mutually statistically independent*, the aim is to determine a matrix $\mathbf{W}$, called the de-mixing matrix, such that $\mathbf{y}_i = \mathbf{W}\mathbf{x}_i$ is an estimate of $\mathbf{s}_i$.

The fundamental approach to solving this problem, is to consider the $\mathbf{y}_i$'s as realizations of a stochastic vector $\mathbf{Y}$, and the $\mathbf{x}_i$'s as realizations of a stochastic vector $\mathbf{X}$. Hence

$$\mathbf{Y} = \mathbf{W}\mathbf{X}. \tag{2.60}$$

The goal is then to iteratively adjust the matrix $\mathbf{W}$ until statistical independence among the random components $Y_1, \ldots, Y_N$, of $\mathbf{Y}$ is reached. In terms of probability densities, this implies that the relation $f(\mathbf{y}) = \prod_{i=1}^{N} f(y_i)$ is obtained.

Several techniques may accomplish this goal. Such techniques are known as *independent component analysis* (ICA). One basic approach is to determine a de-mixing matrix $\mathbf{W}$ such that the mutual information between the components $Y_1, \ldots, Y_N$, is zero, because this means that the components are statistically independent. The learning of $\mathbf{W}$ is accomplished using an iterative procedure, by successively presenting data to the learning system until convergence of the mutual information criterion is reached. Figure 2.5 shows a block diagram of the mutual information ICA setup.

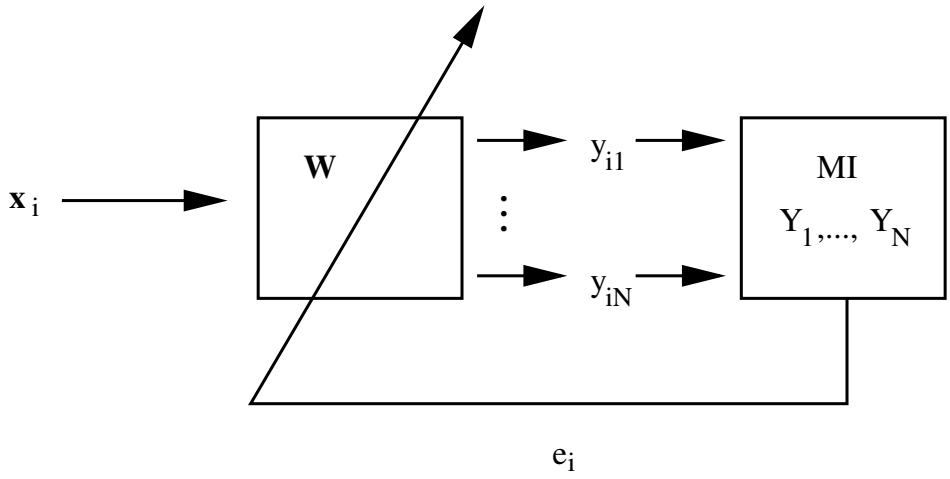


Figure 2.5: *Block diagram of independent component analysis setup.*

In practice, the data is often *prewhitened*, because this preprocessing step makes the process of finding the independent components easier. Prewhitening is accomplished by applying a linear transform $\mathbf{Z} = \mathbf{BX}$, where $\mathbf{B}$ is a $(N \times N)$ matrix, such that the covariance matrix of the random vector $\mathbf{Z}$ equals $E\{\mathbf{ZZ}^T\} = \mathbf{I}$, the identity matrix. Such a procedure removes second order statistical dependence (covariance) in the data, and may be accomplished using principal component analysis (PCA). After prewhitening, what remains is to determine a $(N \times N)$ rotation matrix $\mathbf{R}$, such that the components of $\mathbf{Y} = \mathbf{RZ} = \mathbf{RBX} = \mathbf{WX}$ are as mutually statistically independent as possible. Since the Shannon mutual information can be expressed in terms of Shannon entropies, and since the Shannon entropy is invariant to rotations, we have

$$MI(Y_1, \ldots, Y_N) = \sum_{i=1}^{N} h(Y_i) - h(\mathbf{Y}) = \sum_{i=1}^{N} h(Y_i) - h(\mathbf{Z}) = \sum_{i=1}^{N} h(Y_i), \tag{2.61}$$

since the term $h(\mathbf{Z})$ is independent of the matrix $\mathbf{R}$ and may be disregarded. This property of the Shannon mutual information is important because we avoid having to estimate entropy in high dimensional data spaces, and only need to estimate entropy for one-dimensional signals. Note that the independent components can only be determined up to a permutation and a scaling [Hyvärinen et al., 2001].

Estimation of ICA by minimization of mutual information was probably first proposed by Comon [1994] using an Edgeworth series expansion density estimator. Amari et al. [1996] proposed a related method using a Gram-Charlier series expansion density estimator. A histogram-based approach was derived in [Learned-Miller and Fisher, 2003]. A Parzen window-based algorithm for Kullback-Leibler minimization was derived in [Boscolo et al., 2004], and a Parzen-window-based algorithm for minimizing an approximation of Renyi's mutual information was proposed in [Hild et al., 2001]. Xu et al. [1998] and Erdogmus et al. [2001] proposed related algorithms, and a general Parzen window-based approach in the context of quadratic information measures was put forth in [Principe et al., 2000a]. A clustering-based algorithm for minimizing mutual information was derived in [He et al., 2000]. Pham [1996] proposed a method based on order statistics and Erdogmus et al. [2004b] proposed an algorithm based on Jaynes [1957] maximum entropy principle.

Another basic approach to ICA is given by the principle of non-Gaussianity. Assume that the data has been prewhitened, and that the rotation matrix may be written as

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1^T \\ \vdots \\ \mathbf{r}_N^T \end{bmatrix}, \tag{2.62}$$

where $\mathbf{r}_j^T, j = 1, \ldots, N$, is the jth row of $\mathbf{R}$. Consider Y_j, the jth component of $\mathbf{Y}$. It is given by

$$Y_j = \mathbf{r}_j^T \mathbf{Z} = \mathbf{r}_j^T \mathbf{B}\mathbf{X} = \mathbf{r}_j^T \mathbf{B}\mathbf{A}\mathbf{S}. \tag{2.63}$$

Let $\mathbf{q}_j^T = \mathbf{r}_j^T \mathbf{B}\mathbf{A}$. Hence,

$$Y_j = \mathbf{q}_j^T \mathbf{S} = \sum_{i=1}^{N} q_{ji} S_i, \tag{2.64}$$

a linear combination of the random components $S_1, \ldots, S_N$. The *central limit theorem* states that the distribution of a sum of independent random variables tend toward a Gaussian distribution under certain conditions [Hyvärinen et al., 2001, Delfosse and Loubaton, 1995]. Hence, by determining $\mathbf{q}_j^T$ such that Y_j is maximally *non-Gaussian*, corresponds to a vector having one element $q_{ji} = 1$ and the rest zero, where S_i is the most non-Gaussian random source variable. Hence, the random variable Y_j exactly equals S_i, thus one of the source signals has been determined. This also determines $\mathbf{r}_j^T$. In fact, this approach shows that at most one of the source signals may be Gaussian to begin with, since a sum of two Gaussian variables is a new Gaussian. Therefore, two (or more) Gaussian random variables cannot be separated from each other.

Since the rotation matrix is orthogonal by definition, one may iteratively search for the other non-Gaussian components in directions which are orthogonal to the previously found rows of $\mathbf{R}$. Hence, searching for non-Gaussian components has the great advantage that it allows a deflationary approach, where the independent components are determined one-by-one.

Non-Gaussianity may be measured using cumulant-based measures like kurtosis, which however is very sensitive to outliers. Alternatively, recall that the entropy-based measure called *negentropy*, Eq. (2.12), is a measure of non-Gaussianity. In fact, for uncorrelated, unit variance random variables $Y_1, \ldots, Y_N$, the following relation holds [Hyvärinen et al., 2001]

$$MI(Y_1, \ldots, Y_N) = \text{const.} - \sum_{i=1}^{N} J(Y_i), \tag{2.65}$$

where $J(Y_i)$ is the negentropy of Y_i, and the constant does not depend on the rotation matrix. Hence, minimizing mutual information is equivalent to maximizing the sum of the negentropies, i.e. the non-Gaussianity.

The most famous ICA algorithm which is derived from the principle of non-Gaussianity is the *FastICA* algorithm [Hyvärinen, 1999a, Hyvärinen and Oja, 1997, 1998], which is a fixed-point algorithm which maximizes an approximation of the negentropy measure. As the name suggests, the fixed-point nature of the algorithm makes this method very fast.

Maximum likelihood can also be used to estimate the ICA model. The first approaches to maximum likelihood for ICA were in [Gaeta and Lacoume, 1990, Pham et al., 1992]. The most

well-known such approach is probably the Bell and Sejnowski [1995] algorithm, originally derived from the *infomax* principle, see also [Cardoso, 1997, Obradovic and Deco, 1998]. This algorithm was enhanced in [Amari, 1997], by the introduction of the natural gradient principle. Similar algorithms had already been proposed based on nonlinear decorrelation [Cichocki and Unbehauen, 1996, Cichocki et al., 1994].

We also mention an early approach to ICA, which was based on tensorial methods. The fourth-order cumulants of mixtures are used to define a tensor, which is a generalization of the covariance matrix. The eigenvectors of the tensor more or less directly give the mixing matrix for whitened data. Papers related to tensorial methods are for example [Cardoso, 1989, 1990, Cardoso and Souloumiac, 1993, Comon and Mourrain, 1996]. See for example the book by Hyvärinen et al. [2001] for more on ICA algorithms.

ICA has in the recent years been widely applied in diverse areas like biomedical imaging [Jung et al., 1998, Vigário, 1997, Vigário et al., 1998, 2000], telecommunications [Cristescu et al., 2000a,b], time series prediction [Pawelzik et al., 1996, Eltoft and Kristensen, 2001, Malaroiu et al., 1999] and audio separation [Torkkola, 1999]. See e.g. [Cardoso et al., 1999, Pajunen and Karhunen, 2000, Lee et al., 2001] for more references on ICA applications.

ICA has also been proposed as a generic statistical model for images [Bell and Sejnowski, 1997, Hurri, 1997, Hurri et al., 1997, van Hateren and van der Schaaf, 1998, Hoyer and Hyvärinen, 2000]. In this case an image $\mathbf{x}$ is modeled as:

$$\mathbf{x} = \sum_{i=1}^{N} s_i \, \mathbf{a}_i, \tag{2.66}$$

where $\mathbf{a}_i,\ i = 1, \ldots, N$, are referred to as ICA image basis functions, and $s_i,\ i = 1, \ldots, N$, are statistically independent weighting components.

The ICA basis functions are data dependent in the sense that they are learned from the training data at hand, and they will be different for different training data. This property makes the ICA basis functions fundamentally different from standard linear image representations like Fourier, Haar and cosine transforms. The basis functions can be considered as image building blocks, capturing the inherent features of the training data. ICA may therefore be viewed as a *feature extraction* method in this context. Several authors have applied ICA in order to reveal the features of natural images [Bell and Sejnowski, 1997, Hoyer and Hyvärinen, 2000, van Hateren and van der Schaaf, 1998, Hyvärinen, 1999b], that is, images void of any man made structures. In that case, the training data must be generated from images considered to belong to the class of natural images. When modeling so-called man-made images, images of buildings, cars, etc. are used to generate training data, as in [Hyvärinen, 1999b]. It also turns out that the s_i-components exhibit so-called sparseness, i.e. only a few of the weighting coefficients will have a value significantly deviating from zero. This property is exploited when the model is used in threshold-based image denoising (sparse code shrinkage) [Hyvärinen, 1999b, Jenssen et al., 2001].

In this thesis, we are interested in learning basis images from *textured image data* in order to create a filter bank for segmentation of any textured image sharing the statistical properties of the training data. This is the topic of paper 4.

Chapter 3

Paper 1:
Non-Parametric Clustering by Maximizing the Cauchy-Schwarz PDF Divergence

Non-Parametric Clustering by Maximizing the Cauchy-Schwarz PDF Divergence

Robert Jenssen[1] ROBERTJ@PHYS.UIT.NO
Deniz Erdogmus[2] DERDOGMUS@IEEE.ORG
Kenneth E. Hild II[3] KHILD@CS.UCSF.EDU
Jose C. Principe[4] PRINCIPE@CNEL.UFL.EDU
Torbjørn Eltoft[1] PCTE@PHYS.UIT.NO

1. *Department of Physics*
University of Tromsø, N-9037 Tromsø, Norway

2. *Computer Science and Engineering Department*
Oregon Graduate Institute, OHSU, Portland, OR. 97006, USA

3. *Department of Radiology*
University of San Francisco, San Francisco, CA. 94143, USA

4. *Department of Electrical and Computer Engineering*
University of Florida, Gainesville, FL. 32611, USA

Abstract

In this paper, we develop a practical algorithm for maximizing the recently introduced Cauchy-Schwarz (CS) divergence measure for the purpose of data clustering. The CS divergence in continuous probability spaces can be estimated non-parametrically, and expresses cluster memberships using the Parzen window technique. The actual maximization is carried out by a Lagrange multiplier optimization technique that implements a constrained gradient descent search, with built-in variable step-sizes for each coordinate direction (i.e. no free parameters). The algorithm is also independent of the order of data presentation. To reduce complexity, the gradients are stochastically approximated. An added advantage of the Parzen windowing into the CS maximization is that the risk of convergence to a local optima of the cost function can be reduced by allowing the kernel size to be annealed over a range of values around the optimal value. The optimal value is determined automatically from the data set. We show that the new clustering algorithm is capable of clustering irregularly shaped synthetic data sets, as well as other real data sets, a property we attribute to the information theoretic cost function and the non-parametric estimation of densities.

Keywords: Information theory, Cauchy-Schwarz probability density function distance mesure, Parzen windowing, non-parametric clustering, annealing.

1. Introduction

In pattern recognition and data analysis, it is often desirable to partition, or cluster, a data set into subsets, such that members within subsets are more similar to each other according to some criterion, than to members of other subsets. Clustering is recognized as an important tool in areas such as image segmentation (Frigui and Krishuapuram, 1999), speech recognition (Morgan and Franco, 1997) and signal compression (Abbas and Fahmy, 1994).

Comprehensive introductory texts on traditional clustering algorithms include Duda et al. (2001), Jain and Dubes (1988) and Theodoridis and Koutroumbas (1999). Most of the traditional algorithms, such as K-means (MacQueen, 1967), fuzzy K-means (Bezdek, 1980) and the expectation-maximization algorithm for a Gaussian mixture model (EMGMM) (McLachlan and Peel, 2000), work well for hyper-spherical and hyper-elliptical clusters, since they are often optimized based on a second order statistics criterium. Therefore, in recent years, the main thrust in clustering has been towards developing efficient algorithms capable of handling odd-shaped and highly irregular clusters.

Information-theoretic methods appear as particularly appealing alternatives in this respect, since entropy and pdf distance measures in theory do capture all the information contained in the data distributions in question. The idea of using information theory for clustering is not new. Watanabe (1985) used a coalescence model and a cohesion method to aggregate and shrink the data into desired clusters. Rose et al. (1992) employed the robustness properties of maximum entropy inference for vector quantization, and Hofmann and Buhmann (1997) applied the same criterion for pairwise clustering. Roberts et al. (2000) proposed a clustering method based on minimizing the partition entropy. Recently, Tishby and Slonim (2001) proposed the information bottleneck method. However, all these methods estimate entropy or divergence over discrete spaces for computational simplicity, which is a poor approximation in many cases.

Gokcay and Principe (2002) introduced a Cauchy-Schwarz pdf divergence measure (Principe et al., 2000) as an evaluation metric for clustering and showed that indeed non hyperspherical shaped clusters could be separated with good perfromance. Unfortunately the search algorithm was impractical for large datasets because it required evaluations of distances between pairs of samples. This paper proposes a learning algorithm to maximize the Cauchy-Schwartz pdf divergence as a cost function, and enables practical use of the CS for clustering. One of the difficulties of the task is to find a way to use simple local search procedures in a problem domain that has $0/1$ memberships as clustering. We avoid this difficulty by creating continuous memberships (similarly to fuzzy clustering), which are then adapted in an iterative fashion using the Lagrange multiplier optimization technique. The optimization can be considered a constrained gradient descent search, with built-in variable step-sizes for each coordinate direction. The result is an iterative fixed-point clustering algorithm which has the the advantage of not requiring step-size selection. The only free parameter of the overal procedure is the size of the Parzen kernel, which will be adapted on-line automatically using tools from the statistics literature (Silverman, 1986).

The Parzen kernel size will also be used to avoid a pitfall of gradient descent learning in non-convex cost functions, i.e., the convergence to a local optimum of the cost function. We show that in our algorithm, this problem can be controlled by allowing the size of the

Parzen kernel to be annealed over a range of values around the optimally estimated value. The effect of using a large kernel compared to the optimal kernel size, is to obtain an over-smoothed version of the CS cost function, such that many local optima are eliminated. As the algorithm converges toward the optimum of the smoothed CS divergence, the kernel size is continuously decreased, leading the algorithm toward the true global optimum. We propose a method to select a suitable annealing scheme based on the optimal Parzen kernel, which is, however, rather heuristic at this point.

The original CS divergence fixed-point learning algorithm has a computational complexity of $O(N^2)$, where N is the number of data patterns to be clustered. Thus, it is of crucial importance to reduce the complexity of the algorithm to make it practical for large data sets. To achieve this goal, we derive a stochastic approximation approach to estimate the gradient of CS very much like Widrow's LMS algorithm. The resulting algorithm has complexity $O(MN)$, where M is the number of randomly selected data patterns used in the stochastic approximation. We show that the algorithm performs well even for a very small M, e.g. $M = 0.15N$.

The organization of this paper is as follows. We introduce the Cauchy-Schwarz pdf divergence measure in section 2. In section 3 we derive the CS clustering algorithm, based on the Lagrange multiplier optimization method. Experimental clustering results using the proposed algorithm are presented in section 4. Finally, in section 5, we make our concluding remarks.

2. Cauchy-Schwarz pdf Divergence

Measures of how close two pdfs $p(\mathbf{x})$ and $q(\mathbf{x})$ are in some specific sense, are provided by the stochastic divergences (Kazakos and Papantoni-Kazakos, 1990). There are basically two families of divergence measures, illustrated by the Kullback and Leibler (1951) directed divergence, given by

$$D_{KL}(p,q) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x}, \tag{1}$$

where the stochastic variable $\mathbf{x} \in \mathrm{R}^n$. The other family is the Chernoff divergence given by

$$D_B(p,q) = -\log \int p^{1-t}(\mathbf{x}) q^t(\mathbf{x}) d\mathbf{x}, \quad 0 \leq t \leq 1, \tag{2}$$

where the most well known member is the Bhattacharyya (1943) divergence for $t = \frac{1}{2}$. It is easily shown that both (1) and (2) obey the identity property, that is, they vanish if and only if the two densities are identical (Kazakos and Papantoni-Kazakos, 1990). Note that in order to apply these definitions one needs to estimate the data pdfs, which can be done in a parametric or nonparametric fashion. The problem with the former approach, is that a parametric pdf model must be assumed, such that the parameters of that model can be estimated. Oftentimes, we have no prior knowledge about the shape of the distributions in question. In such situations, the wrong model-choice may give significantly erroneous results. On the other hand, the latter approach makes no assumptions about the parametric form of the distributions, but a pdf estimator is required, such as the Parzen window method. Also, the evaluation of the integrals is normally difficult, hence the conventional

approach of performing the estimations in discrete spaces (integrals become summations). We proceed as follows. Define the inner product between two square-integrable functions, $f(\mathbf{x})$ and $g(\mathbf{x})$, as $\langle f, g \rangle = \int f(\mathbf{x})g(\mathbf{x})d\mathbf{x}$. Then, by the Cauchy-Schwarz inequality,

$$\left| \int f(\mathbf{x})g(\mathbf{x})d\mathbf{x} \right|^2 \leq \int |f(\mathbf{x})|^2 \, d\mathbf{x} \int |g(\mathbf{x})|^2 \, d\mathbf{x}, \tag{3}$$

with equality if and only if the two functions are linearly dependent. Consider again the two pdfs, $p(\mathbf{x})$ and $q(\mathbf{x})$, that is, two strictly non-negative functions. Hence, $\left(\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}\right)^2 \leq \int p^2(\mathbf{x})d\mathbf{x} \int q^2(\mathbf{x})d\mathbf{x}$. A divergence measure between two pdfs can now be expressed as (Principe et al., 2000)

$$D_{CS}(p, q) = -\log J_{CS}(p, q), \tag{4}$$

where

$$J_{CS}(p, q) = \frac{\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}}{\sqrt{\int p^2(\mathbf{x})d\mathbf{x} \int q^2(\mathbf{x})d\mathbf{x}}}. \tag{5}$$

We refer to $D_{CS}(p, q)$ as the Cauchy-Schwarz divergence. We note that $J_{CS}(p, q) \in \langle 0, 1]$, such that $D_{CS}(p, q)$ is always non-negative, it obeys the identity property, and it is also symmetric.

Maximization of D_{CS} is equivalent to minimization of J_{CS}. We will now derive the expression for J_{CS}, by substituting the pdfs with their Parzen estimates.

Parzen windowing is a kernel-based density estimation method, where the resulting density estimate is continuous and differentiable provided that the selected kernel is continuous and differentiable (Devroye and Lugosi, 2001, Parzen, 1962). We are given a set of d-dimensional iid samples belonging to a cluster $C_1 = \{\mathbf{x}_i\}, \ i = 1, \ldots, N_p$, drawn from the density $p(\mathbf{x})$, and the data points $C_2 = \{\mathbf{x}_j\}, \ j = 1, \ldots, N_q$, drawn from $q(\mathbf{x})$. Then, the Parzen window estimators for these distributions are (Parzen, 1962)

$$\begin{aligned} \hat{p}(\mathbf{x}) &= \frac{1}{N_p} \sum_{i=1}^{N_p} W(\mathbf{x} - \mathbf{x}_i), \\ \hat{q}(\mathbf{x}) &= \frac{1}{N_q} \sum_{j=1}^{N_q} W(\mathbf{x} - \mathbf{x}_j), \end{aligned} \tag{6}$$

where W is the Parzen window, or kernel. The Parzen window must integrate to one, and is typically chosen to be a zero mean pdf itself, such as the spherical Gaussian kernel, $W(\mathbf{x} - \mathbf{x}_i) = G(\mathbf{x} - \mathbf{x}_i, \sigma^2\mathbf{I})$, where

$$G(\mathbf{x} - \mathbf{x}_i, \sigma^2\mathbf{I}) = \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} \exp\left\{ -\frac{||\mathbf{x} - \mathbf{x}_i||^2}{2\sigma^2} \right\}. \tag{7}$$

The kernel size is given by the parameter σ^2.

By exploiting an important property of the Gaussian kernel, the resulting expression for the CS divergence can be obtained analytically, without ever explicitly estimating the pdf.

It can be shown that according to the convolution theorem for Gaussians, the following relation holds

$$\int G(\mathbf{x}-\mathbf{x}_i,\sigma^2\mathbf{I})G(\mathbf{x}-\mathbf{x}_j,\sigma^2\mathbf{I})d\mathbf{x} = G_{ij,2\sigma^2\mathbf{I}}, \tag{8}$$

where $G_{ij,2\sigma^2\mathbf{I}} = G(\mathbf{x}_i-\mathbf{x}_j,2\sigma^2\mathbf{I})$.

Thus, when we replace the actual densities in (5) by the Parzen pdf estimates of (6), and utilize (8), we obtain

$$\begin{aligned}\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x} &\approx \int \hat{p}(\mathbf{x})\hat{q}(\mathbf{x})d\mathbf{x}\\ = \frac{1}{N_pN_q}\sum_{i,j=1}^{N_p,N_q}\int G(\mathbf{x}-\mathbf{x}_i,\sigma^2\mathbf{I})G(\mathbf{x}-\mathbf{x}_j,\sigma^2\mathbf{I})d\mathbf{x}&\\ = \frac{1}{N_pN_q}\sum_{i,j=1}^{N_p,N_q}G_{ij,2\sigma^2\mathbf{I}}.&\end{aligned} \tag{9}$$

An exactly similar calculation can be performed for the two quantities in the denominator of (5), yielding

$$\int p^2(\mathbf{x})d\mathbf{x} \approx \frac{1}{N_P^2}\sum_{i,i'=1}^{N_p,N_p}G_{ii',2\sigma^2\mathbf{I}}, \tag{10}$$

and likewise for $\int q^2(\mathbf{x})d\mathbf{x}$, such that

$$\hat{J}_{CS} = \frac{\frac{1}{N_pN_q}\sum_{i,j=1}^{N_p,N_q}G_{ij,2\sigma^2\mathbf{I}}}{\sqrt{\frac{1}{N_p^2}\sum_{i,i'=1}^{N_p,N_p}G_{ii',2\sigma^2\mathbf{I}}\,\frac{1}{N_q^2}\sum_{j,j'=1}^{N_q,N_q}G_{jj',2\sigma^2\mathbf{I}}}}. \tag{11}$$

For each data pattern $\mathbf{x}_i,\ i=1,\ldots,N$, $N=N_p+N_q$, we now define a membership vector $\mathbf{m}_i$. If $\mathbf{x}_i$ belongs to cluster C_1 (C_2), the corresponding crisp membership vector equals $\mathbf{m}_i = [1,0]^T$ ($[0,1]^T$). This allows us to rewrite (11) as a function of the memberships, obtaining

$$\hat{J}_{CS} = \frac{\frac{1}{2}\sum_{i,j=1}^{N,N}\left(1-\mathbf{m}_i^T\mathbf{m}_j\right)G_{ij,2\sigma^2\mathbf{I}}}{\sqrt{\sum_{i,j=1}^{N,N}m_{i1}m_{j1}G_{ij,2\sigma^2\mathbf{I}}\ \sum_{i,j=1}^{N,N}m_{i2}m_{j2}G_{ij,2\sigma^2\mathbf{I}}}}, \tag{12}$$

where m_{ik}, $k=1,2$, denotes element number k of $\mathbf{m}_i$. Note that the variable quantities in (12) are the membership vectors, thus, $\hat{J}_{CS} = \hat{J}_{CS}(\mathbf{m}_1,\ldots,\mathbf{m}_N)$.

In the case of multiple clusters, $C_k,\ k=1,\ldots,K$, we extend the previous definition as follows

$$\hat{J}_{CS} = \frac{\frac{1}{2}\sum_{i,j=1}^{N,N}\left(1-\mathbf{m}_i^T\mathbf{m}_j\right)G_{ij,2\sigma^2\mathbf{I}}}{\sqrt{\prod_{k=1}^{K}\sum_{i,j=1}^{N,N}m_{ik}m_{jk}G_{ij,2\sigma^2\mathbf{I}}}}, \tag{13}$$

where each $\mathbf{m}_i,\ i=1,\ldots,N$, is a binary K dimensional vector. Only the k'th element of any $\mathbf{m}_i$ equals one, which means that the corresponding data pattern $\mathbf{x}_i$ belongs to cluster k.

It is interesting to note that Friedman and Tukey (1974) defined (10) as a measure of the compactness of the cluster. This is in contrast to the Euclidean compactness measure defined by the sum-of-squares error. Moreover, $-\log \int p^2(\mathbf{x})d\mathbf{x}$ is in fact Renyi's quadratic entropy of the random variable described by $p(\mathbf{x})$. Hence, it is clear that the CS divergence and Renyi's entropy are closely related, since the normalizing denominator of (13) takes into account Renyi's entropy of all the clusters. Similarly, (9) can be associated with a "between-cluster entropy".

3. CS Maximization by the Method of Lagrange Multipliers

We are given a data set consisting of the data patterns, $\mathbf{x}_i,\ i = 1, \ldots, N$. Each data pattern is assigned to a crisp membership with respect to the K clusters, represented by the membership vector $\mathbf{m}_i$. Our goal is to assign memberships such that $\hat{J}_{CS}(\mathbf{m}_1, \ldots, \mathbf{m}_N)$ is minimized, because this corresponds to the CS divergence being maximized. We propose to solve this optimization problem by the method of Lagrange multipliers (Rao, 1996). Since this is a technique of differential calculus, we need to make $\hat{J}_{CS}(\mathbf{m}_1, \ldots, \mathbf{m}_N)$ continuous and differentiable. This can be achieved in a variety of ways by smoothing (fuzzyfying) the membership vectors. In fuzzy clustering (Bezdek et al., 1999, Höppner et al., 1999), each data point is no longer restricted to be assigned to only one cluster, as in crisp clustering, but may be assigned to several clusters at the same time. Fuzzy K-means (Bezdek, 1980) is perhaps the most well-known such clustering algorithm. Here, however, the membership function does not play as crucial a role as in fuzzy clustering, because it is solely used as a regularizer of the membership value during adaptation.

Our first step is to let the elements of $\mathbf{m}_i$ have values in the interval $[0, 1]$, for all $i = 1, \ldots, N$. Now we define the following constrained optimization problem:

$$\min_{\mathbf{m}_1, \ldots, \mathbf{m}_N} \hat{J}_{CS}(\mathbf{m}_1, \ldots, \mathbf{m}_N), \tag{14}$$

subject to

- $\mathbf{m}_j^T \mathbf{1} - 1 = 0, \quad j = 1, \ldots, N,$

where $\mathbf{1}$ is a K-dimensional vector whose elements are all one. Hence, a data pattern is allowed to have a certain degree of membership to any cluster, but the constraint ensures that the sum of the memberships adds up to one.

Now, we make a change of variables which allows us to derive a fixed-point, gradient-based, learning rule for clustering, which requires no step size to be selected. Let $m_{ik} = v_{ik}^2,\ k = 1, \ldots, K$. Consider the following optimization problem

$$\min_{\mathbf{v}_1, \ldots, \mathbf{v}_N} \hat{J}_{CS}(\mathbf{v}_1, \ldots, \mathbf{v}_N), \tag{15}$$

subject to

- $\mathbf{v}_j^T \mathbf{v}_j - 1 = 0, \quad j = 1, \ldots, N.$

The constraints for the problem stated in (15) with regard to $\mathbf{v}_j,\ j = 1, \ldots, N$, are equivalent to the constraints for the problem stated in (14) with regard to $\mathbf{m}_j,\ j = 1, \ldots, N$. Note

that given vectors $\mathbf{v}_j,\ j = 1, \ldots, N$, obeying (15), it follows that the elements of the corresponding $\mathbf{m}_j \in [0, 1]$. The optimization problem, (15), amounts to adapting the vectors $\mathbf{v}_i,\ i = 1, \ldots, N$, such that

$$\frac{\partial \hat{J}_{CS}}{\partial \mathbf{v}_i} = \left(\frac{\partial \hat{J}_{CS}}{\partial \mathbf{m}_i}^T \frac{\partial \mathbf{m}_i}{\partial \mathbf{v}_i} \right)^T = \mathbf{\Gamma} \frac{\partial \hat{J}_{CS}}{\partial \mathbf{m}_i} \rightarrow \mathbf{0}, \tag{16}$$

where $\mathbf{\Gamma} = \text{diag}(2\sqrt{m_{i1}}, \ldots, 2\sqrt{m_{iK}})$. Notice that if all of the diagonal elements $2\sqrt{m_{ik}},\ k = 1, \ldots, K$, are positive, the direction of the gradients of $\frac{\partial \hat{J}_{CS}}{\partial \mathbf{v}_i}$ and $\frac{\partial \hat{J}_{CS}}{\partial \mathbf{m}_i}$ will always be the same. Hence, in this case, these scalars can be thought of as variable step sizes built into the gradient descent search process, as a consequence of the change of variables that we made. We force all the elements of the membership vectors $\mathbf{m}_i,\ i = 1, \ldots, N$, to always be positive, by adding a small positive constant ϵ (e.g. $\epsilon \sim 0.05$) to all the elements during each membership update in the iterative algorithm based on the method of Lagrange multipliers. The actual value of ϵ is not critical for our purpose, as long as it is a small number, since it's only task is to keep the memberships from ever going to zero. This approach also has the effect of introducing a small amount of noise into the algorithm, a strategy that is well-known as an additional means to help avoid local optima. See the Appendix for the derivation of $\frac{\partial \hat{J}_{CS}}{\partial \mathbf{m}_i}$.

The necessary conditions that the solution of (15) must obey, are commonly generated by constructing a function, $L = L(\mathbf{v}_1, \ldots, \mathbf{v}_N, \lambda_1, \ldots, \lambda_N)$, known as the Lagrange function (Rao, 1996), given by

$$L = \hat{J}_{CS}(\mathbf{v}_1, \ldots, \mathbf{v}_N) + \sum_{j=1}^{N} \lambda_j \left(\mathbf{v}_j^T \mathbf{v}_j - 1 \right), \tag{17}$$

where $\lambda_j,\ j = 1, \ldots, N$, are the *Lagrange multipliers*. The necessary conditions for the extremum of L, which also corresponds to the solution of the original problem, (15), are given by

$$\frac{\partial L}{\partial \mathbf{v}_i} = \frac{\partial \hat{J}_{CS}}{\partial \mathbf{v}_i} + \sum_{k=1}^{N} \lambda_k \frac{\partial}{\partial \mathbf{v}_i} \left(\mathbf{v}_k^T \mathbf{v}_k - 1 \right) = \mathbf{0}, \tag{18}$$

$$\frac{\partial L}{\partial \lambda_j} = \mathbf{v}_j^T \mathbf{v}_j - 1 = 0, \tag{19}$$

for $i = 1, \ldots, N$ and $j = 1, \ldots, N$. From (18) we derive the following *fixed-point* adaption rule for the vector $\mathbf{v}_i$ as follows

$$\begin{aligned} &\frac{\partial \hat{J}_{CS}}{\partial \mathbf{v}_i} + 2\lambda_i \mathbf{v}_i = \mathbf{0}, \\ &\Rightarrow \mathbf{v}_i^+ = -\frac{1}{2\lambda_i} \frac{\partial \hat{J}_{CS}}{\partial \mathbf{v}_i}, \end{aligned} \tag{20}$$

$i = 1, \ldots, N$, and where $\mathbf{v}_i^+$ denotes the updated vector.

We solve for the Lagrange multipliers, $\lambda_i,\ i = 1, \ldots, N$, by evaluating the constraints given by (19) as follows

$$\begin{aligned} & \mathbf{v}_i^{+T}\mathbf{v}_i^{+} - 1 = 0, \\ \Rightarrow & \left(-\frac{1}{2\lambda_i}\,\frac{\partial \hat{J}_{CS}}{\partial \mathbf{v}_i}\right)^T \left(-\frac{1}{2\lambda_i}\,\frac{\partial \hat{J}_{CS}}{\partial \mathbf{v}_i}\right) - 1 = 0, \\ & \Rightarrow \lambda_i = \frac{1}{2}\sqrt{\frac{\partial \hat{J}_{CS}}{\partial \mathbf{v}_i}^T \frac{\partial \hat{J}_{CS}}{\partial \mathbf{v}_i}}. \end{aligned} \tag{21}$$

After convergence of the algorithm, or after a predetermined number of iterations, we designate the maximum value of the elements of each membership vector $\mathbf{m}_i,\ i = 1, \ldots, N$, to one, and the rest to zero.

We initialize the membership vectors randomly according to a uniform distribution. That way $\mathbf{m}_i \in [0, 1]\ \forall i$, even though the constraint of (14) is not obeyed. We have observed that after the first iteration through the algorithm, the constraint is always obeyed. Better initialization schemes may be used, although in our experiments, the algorithm is very little affected by the actual initialization used.

Note that the only free parameters of the CS clustering algorithm derived above, is the kernel size in the CS divergence, σ^2, and the number, K, of clusters. In the following subsections, we show that the kernel size, which is fundamentally linked to pdf estimation, can be determined automatically based on the data at hand by employing tools from the density estimation literature. We will also discuss annealing of the kernel as a learning strategy to help avoid convergence to a local minimum of the cost function. Furthermore, we also show how the computational complexity of the algorithm can be successfully reduced, and discuss the problem of determining the number of clusters.

3.1 Choice of kernel size

Many approaches have been proposed in order to optimally determine the size of the Parzen window, given a finite sample data set. Silverman (1986) discussed this problem, using the asymptotic mean integrated square error (AMISE) between the estimated an the actual pdf as the optimality metric. However, the optimal kernel size based on AMISE, depends on the second derivative of the actual pdf itself, hence it is not practical. But by estimating the unknown quantity as if the underlying density is Gaussian, it can be plugged back into the expression for the optimal kernel size, yielding Silverman's rule-of-thumb criterion for d-dimensional data

$$\sigma_{\text{opt}} = \sigma_X \left\{4N^{-1}(2d+1)^{-1}\right\}^{\frac{1}{d+4}}, \tag{22}$$

where $\sigma_X^2 = d^{-1}\sum_i \Sigma_{X_{ii}}$ and $\Sigma_{X_{ii}}$ are the diagonal elements of the sample covariance matrix. More advanced kernel size selection techniques are reviewed in (Jones et al., 1996, Duin, 1976, Schraudolph, 2004). Note that although we have derived the CS clustering algorithm using a single spherical Parzen kernel, Parzen estimators based on a optimizing a diagonal covariance matrix, or a full covariance matrix, exist, and can also be utilized in the CS algorithm.

There is no guarantee that a single spherical Parzen kernel is appropriate in order to estimate the pdf of each cluster, but our experiments demonstrate this choice to be sufficient to reveal the clusters. In our concluding remarks, we will discuss how the algorithm may be modified in order to learn a separate kernel size for each cluster as the iterative algorithm proceeds.

3.2 Kernel size annealing

Solving clustering using local search is subject to many local minima due to the nonconvex nature of the cost function, therefore practical algorithms must address this issue. One of the advantages of our algorithm is that the Parzen kernel size can be annealed and used effectively as convolution smoothing. We show experimentally that in our algorithm local minima can to a high degree be avoided, by allowing the size of the kernel to be annealed over an interval of values around the optimal value. The Parzen estimator is biased since the expected value of, for example $\hat{p}(\mathbf{x})$, is $E\{\hat{p}(\mathbf{x})\} = p(\mathbf{x}) * G(\mathbf{x}, \sigma^2\mathbf{I})$, where $*$ denotes convolution. Thus, if the kernel size, σ^2, is too large, $E\{\hat{p}(\mathbf{x})\}$ is an over-smoothed version of $p(\mathbf{x})$. The bias can be asymptotically reduced to zero by reducing the kernel size monotonically with increasing number of samples, so that the kernel asymptotically approaches a dirac-delta function. In the finite sample case, the kernel size can not be zero and has to be chosen in a trade-off between estimation bias and variance, since a small kernel size increases the variance of the estimator. In our optimization, since the estimated CS divergence depends explicitly on the Parzen estimates, it is likely that the estimated CS divergence will be an over-smoothed version of the actual CS divergence. Hence, the estimated CS divergence using a "large" kernel is likely to be smoother, and thus exhibit fewer local minima, than the estimated CS divergence using a "small"kernel. At the same time, the global minimum of the over-smoothed CS estimate is likely to be biased with respect to the global minimum of the actual CS divergence. The approach taken, is to let the algorithm iterate toward the minimum of the over-smoothed CS divergence, while continuously decreasing the kernel size, hence leading the algorithm toward the actual global optimum. This approach resembles the annealing of the temperature parameter in simulated annealing (Kirkpatrick et al., 1983), and to the theory of convolution smoothing for optimization (Styblinski and Tang, 1990, Erdogmus and Principe, 2002). We will return to this issue in section 4, and also propose a method to select the upper limit for the kernel size which is, however, rather heuristic.

3.3 Reducing complexity

The fixed-point algorithm we have derived is order independent, i.e. the order in which the data patterns are presented to the algorithm is of no importance.

However, the computation of all the gradients $\frac{\partial \hat{J}_{CS}}{\partial \mathbf{m}_i}$, $i = 1, \ldots, N$, is an $O(N^2)$ procedure. In practical clustering problems, the data sets may be very large. Thus, it is of crucial importance to reduce the complexity of the algorithm. The expression for the gradient $\frac{\partial \hat{J}_{CS}}{\partial \mathbf{m}_i}$ is derived in the Appendix. The key point to note, is that we can calculate all quantities of interest in (27), by determining (28), for $\forall i$. Since (28) is a sum over N elements, calculating all these quantities is an $O(N^2)$ procedure. To reduce complexity,

we estimate (28) by *stochastically sampling* the membership space, and utilize M randomly selected membership vectors, and corresponding data points, to compute

$$\sum_{m=1}^{M} \mathbf{m}_m G_{im,2\sigma^2\mathbf{I}}, \tag{23}$$

as an approximation to (28). Hence, the overall complexity of the algorithm is reduced to $O(MN)$ for each iteration. We will show that we obtain very good clustering results, even for very small M, e.g. $M = 0.15N$.

3.4 The number of clusters

In many cases, the desired number of clusters, K, in a data set is not known beforehand. Determining the number of clusters is a fundamental problem in exploratory data analysis and clustering. There exists no universally accepted procedure to solve this task for arbitrary data sets. For example, methods based on intra-cluster distances, are not necessarily appropriate in our case, since the data sets may have irregular shapes. Many traditional cluster validity criteria are reviewed in (Jain and Dubes, 1988). It is a fact that most existing clustering schemes which rely on the optimization of a cost function using differential calculus techniques assume a-priori knowledge about the number of clusters (Theodoridis and Koutroumbas, 1999). When the number of clusters are not known beforehand, these algorithms are often executed for several K-values, for thereafter to determine K based on the value of the cost function in each case. Such a search for the correct K can be computationally very demanding. The CS clustering algorithm also needs K to be specified.

One possible method to estimate K for our purposes is the following. As we have noted previously, the CS cost function is in fact closely related to the Renyi entropy of the clusters. Recently, Girolami (2002) proposed a method for estimating the number of clusters in a data set based on Renyi's entropy of the data and the eigendecomposition of a kernel matrix. As noted in (Girolami, 2002), the elements $G_{ij,2\sigma_2\mathbf{I}},\ i,j = 1,\ldots,N$, can be interpreted as the elements $A_{ij},\ i,j = 1,\ldots,N$, of an affinity, or kernel matrix $\mathbf{A}$. This matrix can be diagonalized as $\mathbf{\Lambda} = \mathbf{E}^T\mathbf{A}\mathbf{E}$, where the columns, $\mathbf{e}_1,\ldots,\mathbf{e}_N$, of $\mathbf{E}$ contain the eigenvectors of $\mathbf{A}$, and $\mathbf{\Lambda}$ is a diagonal matrix with the associated eigenvalues $\lambda_1,\ldots,\lambda_N$. Let $f(\mathbf{x})$ be the pdf of the whole data set. Then Renyi's quadratic entropy can be estimated by the Parzen method as (Principe et al., 2000)

$$H_{R_2}(\mathbf{x}) = -\log V(\mathbf{x}), \tag{24}$$

where $V(\mathbf{x})$ can be written (Girolami, 2002)

$$\begin{aligned} V(\mathbf{x}) &= \frac{1}{N^2}\sum_{i,j=1}^{N,N} A_{ij}, \\ &= \frac{1}{N^2}\sum_{i=1}^{N} \lambda_i\{\mathbf{1}^T\mathbf{e}_i\}, \end{aligned} \tag{25}$$

where $\mathbf{1}$ is a N-dimensional all ones vector. Girolami showed for several datasets that if there are K distinct clustered regions within the N data samples then there will be K

dominant terms $\lambda_i\{\mathbf{1}^T\mathbf{e}_i\}$ in the summation (25). In our experiments, we assume a-priori knowledge about the number of the clusters, also in order to be able to compare clustering algorithms. But we note that for practical use of the CS algorithm, the above mentioned method for estimating K may be the most promising.

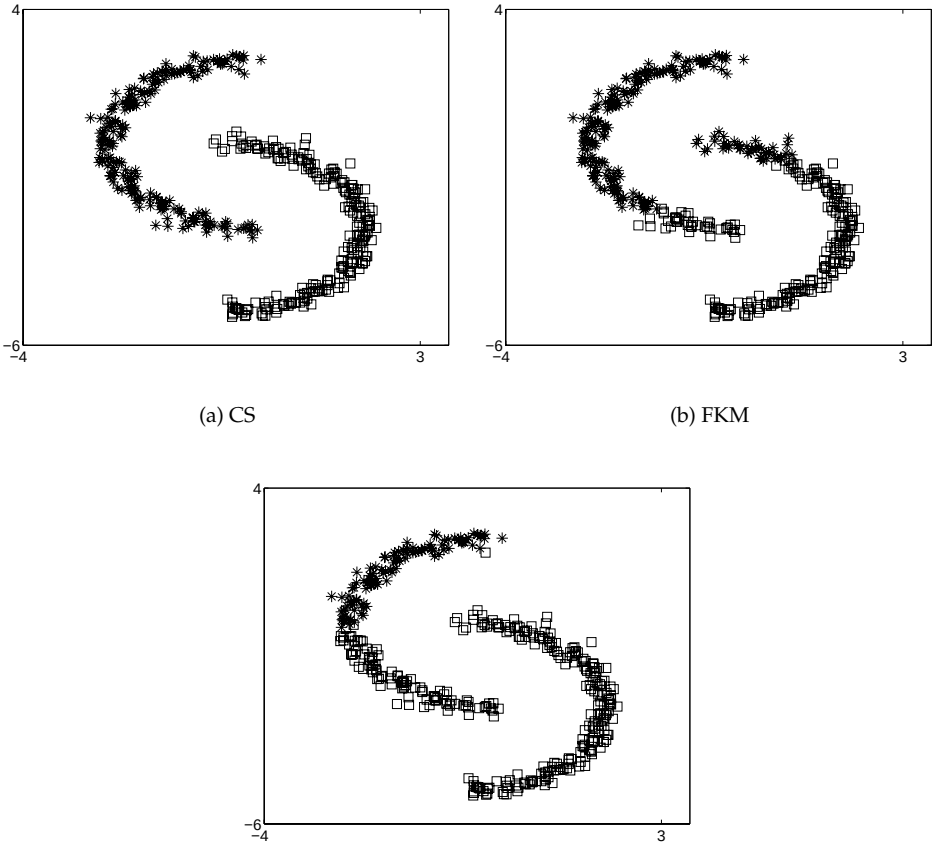
4. Results

In this section, we demonstrate that the CS clustering algorithm is able to unravel the underlying structure of several different data sets, and hence to produce reasonable clusters. For comparison, we also show the results obtained for fuzzy K-means (FKM) (Bezdek, 1980), the EMGMM algorithm (McLachlan and Peel, 2000) and the single-link (SL) hierarchical clustering algorithm. These methods are all popular clustering algorithms.

The fuzzy K-means algorithm (Bezdek, 1980) optimizes a cost function based on the sum-of-squares distance between the cluster centroids and the data points. It is therefore based on a variance criterion, and creates a linear cluster boundary. It is in frequent use, also since it has a low computational complexity of only $O(N)$. The EMGMM algorithm (McLachlan and Peel, 2000) models the data as a mixture of Gaussian distributions, and performs the optimization by means of the expectation-maximization method. In its most advanced mode of operation, when optimizing the full covariance structure of the Gaussians, it requires the inversion of K covariance matrices at each iteration step, and thus has a complexity of $O(Kd^2N)$. The single-link algorithm (Duda et al., 2001) starts out with N clusters, and joins clusters in a repeated process, based on the distance between the nearest inter-cluster data points. As long as the clusters are dense, the single-link algorithm can handle many odd-shaped data sets.

For all algorithms, the correct number of clusters is provided. For the CS algorithm, the memberships are initialized randomly, as suggested in section 3, and the constant ϵ is set to 0.05. For all experiments, we utilize 15% of the data set when estimating the gradients used in the CS update rule.

The CS clustering algorithm is illustrated in two modes of operation. We conduct experiments using both a fixed kernel, and an annealing scheme for the kernel size. For a fixed kernel, the optimal kernel size, σ_{opt}, is determined by (22), according to the MISE criterion. Hence, there are no user specified parameters, making the algorithm fully automatic. When annealing, the upper limit for the kernel size is set to $\sigma_{\text{upper}} = 2\sigma_{\text{opt}}$ and the lower limit to $\sigma_{\text{lower}} = 0.5\sigma_{\text{opt}}$. The kernel size is linearly decreased using a step size $\Delta_\sigma = (\sigma_{\text{upper}} - \sigma_{\text{lower}})/100$. If convergence is not obtained when reaching σ_{lower}, the algorithm continues in fixed kernel mode using σ_{lower} as the kernel size. The choice of the annealing scheme is based on our experience. We have found that $\sigma_{\text{upper}} = 2\sigma_{\text{opt}}$ annealed over 100 iterations always gives good results in the experiments we have conducted, especially on low dimensional data. For the real data sets that we examine in this study, which have higher dimensionality, we have found that the performance of the algorithm may improve slightly by selecting $\sigma_{\text{upper}} = 3\sigma_{\text{opt}}$. Selecting σ_{upper} any larger than these values, only has the effect that the algorithm needs a slower annealing scheme, and hence more iterations for convergence. The lower limit is set to be smaller than σ_{opt} since the MISE criterion is known to output a kernel size which is truly optimal only with respect



(a) CS

(b) FKM

(c) EMGMM

Figure 1: Odd-shaped data set. MISE optimal kernel size is $\sigma_{\text{opt}} = 0.66$.

to a Gaussian distribution, and may produce a too smooth pdf estimate for non-Gaussian data.

The convergence criterion may be determined based on the value of the CS cost function, on the change in fuzzy memberships, or based on the change in crisp memberships. In this paper, we examine the crisp memberships every tenth iteration. If there is no change in crisp memberships over these ten iterations, it is assumed that the algorithm has either converged to a reasonable solution, or that the algorithm is trapped in a local minimum from which it cannot escape. Hence, when the algorithm terminates, it has in practice converged at least ten iterations earlier.

We will conduct experiments on four synthetic data sets and two real. The synthetic data sets are referred to as SET 1, SET 2, SET 3 and SET 4. They are limited to two-dimensional feature spaces to make it easy to visualize the data assignments. SET 1 con-

sists of two clusters and contains a total of $N = 419$ data patterns. The data set is shaped as two half-circles with highly non-linear cluster boundaries. It is a very difficult data set for the traditional algorithms to handle. The purpose is to show that the CS algorithm can handle this odd-shaped data set. SET 2 is actually the same type of data as SET 1, and contains the same number of data patterns. However, while the clusters of SET 1 are dense, the clusters in SET 2 is much more noisy. The cluster boundaries are therefore much less clear. The purpose is to show that the CS algorithm performs well on an odd-shaped noisy data set. The purpose of including SET 3, is to show that the CS algorithm can cluster irregularly shaped data very fast, also compared to successful clustering by the EMGMM algorithm. SET 3 contains $N = 796$ data pattern. It consists of one Gaussian cluster partially surrounded by a half-circle shaped cluster. The clustering of SET 4 shows that the CS algorithm can handle odd-shaped clusters, also when some of the clusters are dense, and others more sparse. It consists of two dense Gaussian clusters and one string of data points, shaped almost as the letter 's'. It has $N = 526$ data points. We cluster each data set 20 times in each mode in order to study the stability properties of the algorithm. We want to quantify the percentage of times it converges to a local minimum contra the global minimum of the cost function.

The real data sets are the Wisconsin-breast cancer (WBC) data (Mangasarian and Wolberg, 1990) and a texture segmentation data set. They have been included to show that the CS clustering algorithm also work well for real data in high dimensions. The WBC data set is extracted from the UCI repository data base (Murphy and Ada, 1994). The texture segmentation data set is obtained by filtering a textured image through a bank of Gabor filters, from which features appropriate for texture segmentation can be generated.

4.1 Synthetic data

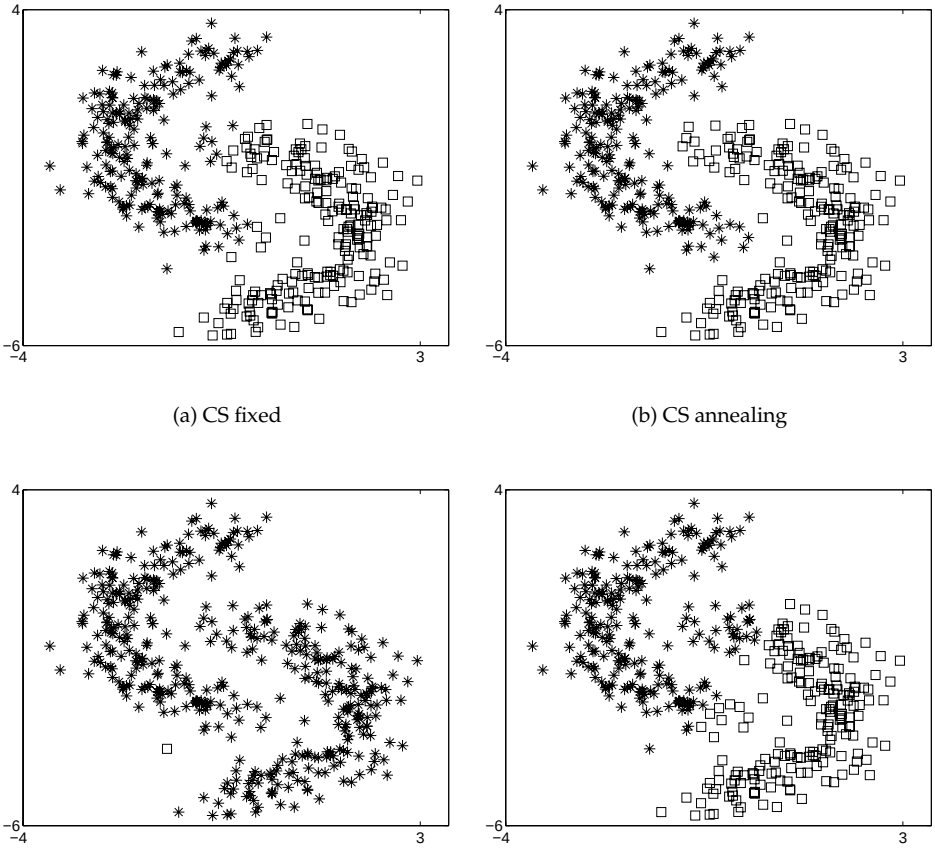
SET 1. For this data set, shown in Fig. 1, the optimal kernel size was determined to be $\sigma_{\text{opt}} = 0.66$. Using the algorithm in the fixed kernel mode, the result shown in Fig. 1 (a) was obtained in 75% of the trials. Visually, it is clear that this is a clustering which reflects the underlying structure of the data set, as determined by a human observer. In many of the trials, this result was obtained in as little as 10-30 iterations. In the annealing mode, the CS algorithm in fact produced the result shown in Fig. 1 (a) in every single trial. We attribute this property to the smoothing effect inherent in the algorithm when operated in this mode, which helps avoid local minima.

SET 1 is favorable to the single-link algorithm since the clusters are dense. Hence, in this case, the single-link algorithm obtains the same result as the CS clustering algorithm, which is the correct solution.

The best result using Fuzzy K-means is shown in Fig. 1 (b). It creates a linear cluster boundary, which is clearly wrong. The cluster centroids were initialized randomly by drawing from the data set.

A typical result of using the EMGMM algorithm is shown in Fig. 1 (c). The EMGMM algorithm also fails in every trial, even when optimizing the full covariance. The cluster centroids were initialized as the centroids determined by the K-means algorithm.

SET 2. The SET 2 data set is shown in Fig. 2. The kernel size is determined to be $\sigma_{\text{opt}} = 0.67$. We find that the performance of the CS algorithm is quite robust to the presence



(a) CS fixed

(b) CS annealing

(c) SL

(d) EMGMM

Figure 2: Noisy odd-shaped data set. MISE optimal kernel size is $\sigma_{\text{opt}} = 0.67$.

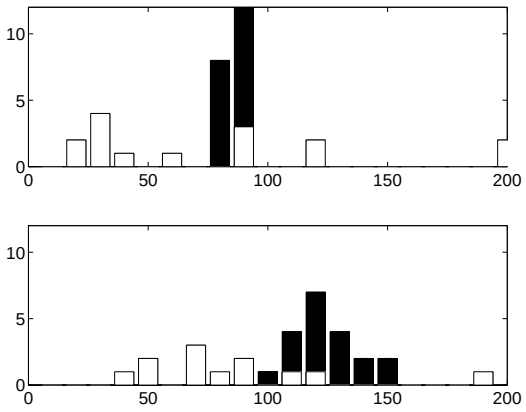


Figure 3: Distribution of stopping times for SET 1 (upper panel) and SET 2 (lower panel). The white bars indicate the number of times the algorithm stopped at a certain iteration step in fixed kernel mode, while the black bars correspond to the annealing mode.

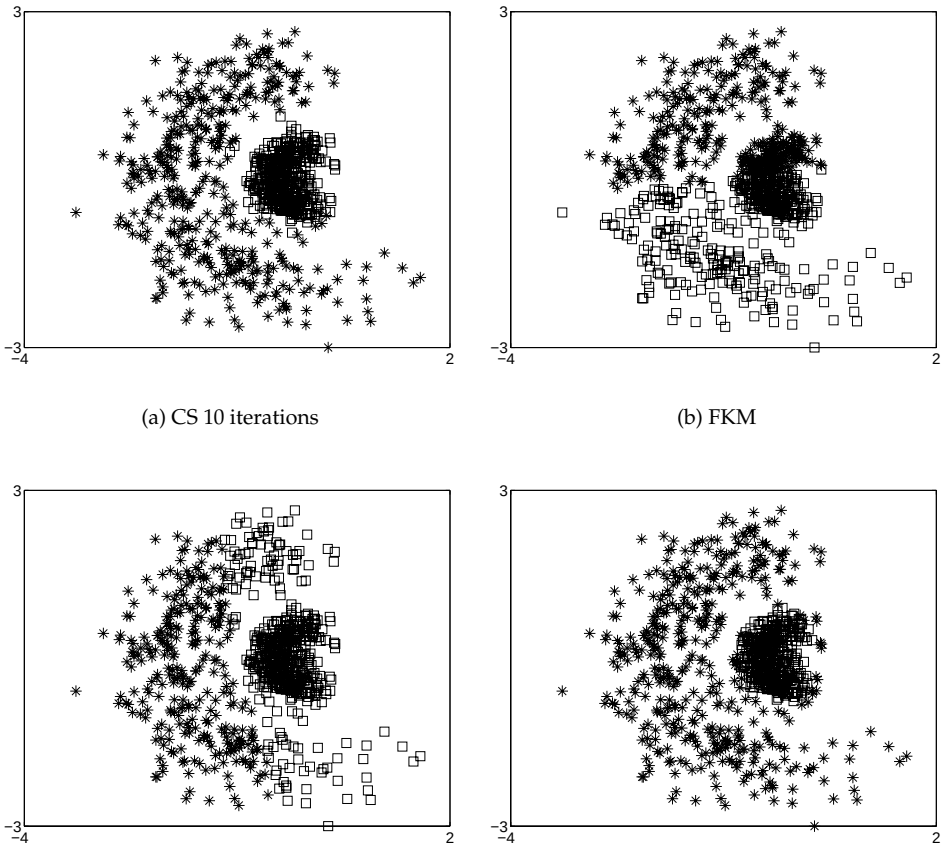
of noise, but that it converges to a local minimum more often. In 60% of the fixed mode trials the CS algorithm produces results in agreement with a visual clustering. A typical result is shown in Fig. 2 (a). It can be seen that the algorithm produces non-linear cluster boundaries, but has some problems in the most difficult regions. This indicates that σ_{opt} is a little bit too large for a perfect result. However, when operated in the annealing mode, the algorithm converges to the result shown in Fig. 2 (b) in absolutely all trials.

Fig. 2 (c) shows the result obtained by the single-link method. The result is indicative of the main problem of this algorithm. It may break down completely when the data is noisy and there is overlap between the clusters. This is because the wedge points have the effect of erroneously linking together clusters. Fortunately, the CS clustering algorithm shows no such tendency.

The best result of using the EMGMM algorithm is shown in Fig. 2 (d). The full covariance structure has been optimized. It is clear that this algorithm cannot handle this data set very well. Using the same covariance for all clusters produces a significantly poorer result.

The result for fuzzy K-means has not been shown, because it produces essentially the same result as the EMGMM algorithm.

Annealing analysis: SET 1 and SET 2. We now discuss the annealing property in more detail. Fig. 3 shows the distribution of stopping times for the CS algorithm. The white bars in the upper panel show the number of times the algorithm stopped at a certain iteration step, for those 75% of the trials when the result shown in Fig. 1 (a) was obtained. It can be seen that in many of the trials the algorithm stopped after only 20-40 iterations. But in some trials, the convergence was slow, and in two cases even 200 iterations was needed. In these cases, the algorithm got of to a bad start, and only managed to converge toward the



(a) CS 10 iterations

(b) FKM

(c) EMGMM typical

(d) EMGMM best

Figure 4: Data set with non-linear cluster boundary. The CS algorithm is able to cluster this data set correctly in a few iterations.

global minimum after a very long time. Also, in 25% of the trials, the algorithm converged to a local solution, and produced a final result similar to that of Fig. 1 (b). However, when annealing, the algorithm converged to the result shown in Fig. 1 (a) in every single trial. Moreover, it always stopped after either 80 or 90 iterations, as indicated by the black bars. That is, the algorithm was somewhat slower than in the best fixed kernel trials, but did always converge to the global minimum. In practice, the algorithm always moved very rapidly to a fuzzy K-means like, linear solution, as in Fig. 1 (b), for thereafter to converge more slowly towards the correct non-linear solution as the kernel size decreased. This kind of behaviour seems to be typical of the CS clustering algorithm in the annealing mode.

The lower panel of Fig. 3 shows the same kind of distribution for the stopping times for the SET 2 data. It can be seen that in the fixed kernel mode, the stopping times for those 60% of the trials which produced a result similar to that shown in Fig. 2 (a), are more spread than in the previous case, and are also higher on average. The same tendency is clear also when the algorithm operates in annealing mode. But again, in annealing mode, the algorithm always converged to the global solution, in this case to the result shown in Fig. 2 (b). In practical applications, the robustness of the CS algorithm with regard to avoiding convergence to a local minimum may be very important.

SET 3. Fig. 4 (a) shows a typical result obtained by the CS algorithm in the fixed kernel mode. It reached such a result in 75% of the trials. In most of these trials, this result was obtained in just a few iterations, often as low as 10 iterations. This shows that the CS algorithm may converge quickly. Again, in annealing mode, the algorithm always converges to a global minimum, but slower, on average in about 70 iterations. For illustration, Fig. 4 (b) shows that fuzzy K-means fails. When optimizing the full covariance structure of the Gaussians, the EMGMM algorithm produced a result similar to that shown in Fig. 4 (c) in about 75% of the trials. In the remaining trials, it produced a result similar to that of Fig. 4 (d), which is almost identical to the CS clustering result. When the EMGMM algorithm did converge to this solution, it did so in about 40 iterations (in our experiments, the algorithm terminates when the difference in log-likelihood between two consecutive iterations is less than 10^{-4}).

SET 4. Also for SET 4, shown in Fig. 5, the CS algorithm produced a perfect result in about 75% of the trials when operated in fixed kernel mode. Once again, by annealing, the clustering is consistently robust, but slower.

This data set is easily clustered by the single-link algorithm, while fuzzy K-means fails. The EMGMM algorithm occasionally succeeded, in a about 20% of the trials.

4.2 Breast-cancer data

In this experiment, we cluster breast-cancer data into the two classes *benign* and *malignant*. The Wisconsin Breast-Cancer (WBC) database (Mangasarian and Wolberg, 1990) is the source of this dataset, which consists of 683 data points (444 benign and 239 malignant). WBC is a nine-dimensional dataset with the following features: i) Clump thickness; ii) Uniformity of cell size; iii) Uniformity of cell shape; iv) Marginal adhesion; v) Single epithelial cell size; vi) Bare nucli; vii) Bland chromatin; viii) Normal nucleoli; and ix) Mitoses.

To ensure convergence to a reasonable solution for the real data sets, the CS algorithm is operated in annealing mode, using the annealing scheme discussed in the beginning of

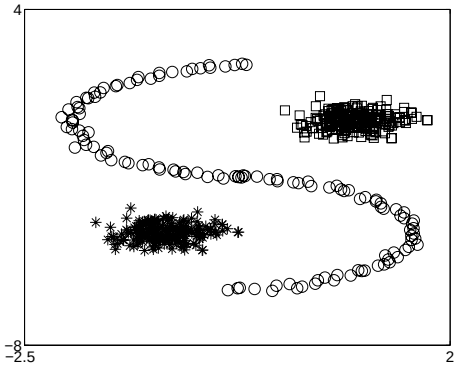


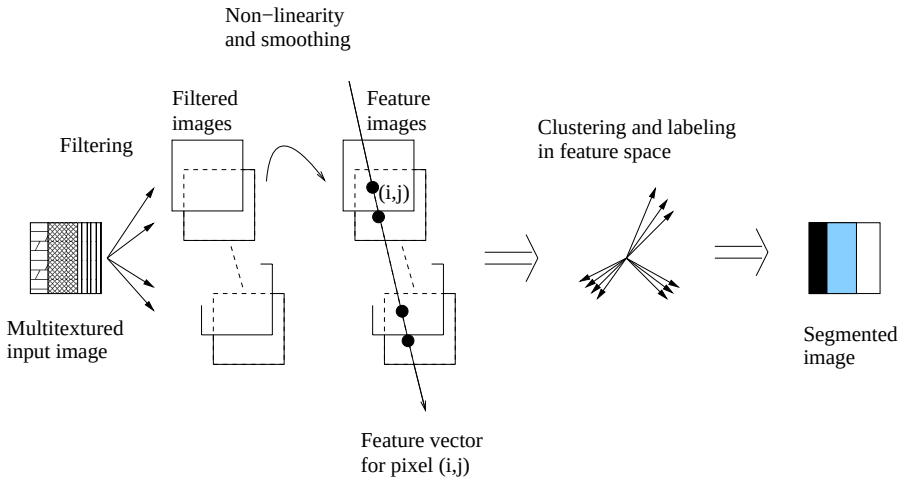
Figure 5: Data set consisting of two dense Gaussian clusters and one irregular string of data points. The CS algorithm is able to cluster it correctly.

this section. The CS clustering algorithm obtained a classification accuracy of 94.4% on average. The algorithm stopped after about 50 iterations on average. This result is almost identical to the result obtained by the EMGMM algorithm, which was 94.5% on average. However, for the EMGMM algorithm this result was only obtained after the algorithm iterated over the predefined maximum number of iterations, which is 1000 in this case. For example, using 300 iterations, the EMGMM algorithm only obtained an accuracy of 90.2% on average. For the WBC data set, fuzzy K-means in fact performed slightly better than the CS algorithm. It obtained an accuracy of 95.5% on average. This means that the WBC data set can be clustered reasonably well by creating linear cluster boundaries.

4.3 Texture segmentation

As a final experiment, we utilize the CS clustering algorithm in the task of segmenting a multi-textured image into its distinct regions. Texture segmentation consists of three fundamental steps. 1) Generate a feature vector corresponding to each pixel location. 2) Cluster and label the feature vectors by some clustering algorithm. 3) Create the segmented image, where the intensity value of each pixel is the label of the corresponding feature vector. We need to briefly discuss the first step before proceeding.

A popular method for generating features appropriate for texture segmentation, is to filter a composite textured image through a bank of filters, and to generate features based on the filter outputs. The features are usually based on the energy of the filtered images. The energy of the filtered images depends on the orientation properties and frequency characteristics of the filters, and on the match between these and the various textured regions. That is, the filter output for textured regions with properties similar to the filter itself, tends to have high energy. Conversely, for regions where the texture and the filter do not match, the filter output corresponding to that region tends to have low energy. The regions of low and high energy in the filtered images can be identified by applying a



(a)

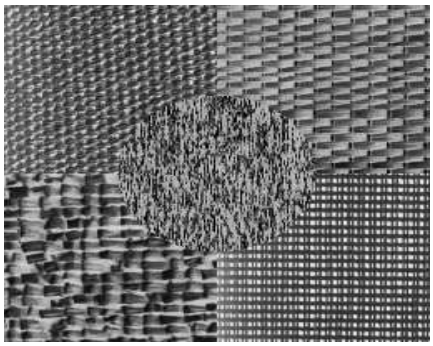
Figure 6: Schematic illustration of experimental setup for texture segmentation.

non-linearity, such as squaring, followed by a smoothing operation to the filtered images. The resulting energy images are used to generate the features. The experimental setup described above for texture segmentation is shown in Fig. 6.

Jain and Farrokhnia (1991) used a dyadic even-symmetric Gabor filter bank to generate the energy features. We follow their approach in this experiment. Gabor filters can be interpreted as oriented bandpass filters. The Gabor filters used in (Jain and Farrokhnia, 1991) were oriented along the horizontal, vertical and two diagonals. For a (256×256) image, five radial center frequencies were used, all separated by one octave, such that the pass-band of the filter with the highest frequency falls below the Nyquist frequency. Hence, there are 20 filters in the filter bank, and the resulting data set to be clustered is 20-dimensional. Since there are 256^2 pixels, the data set is very large. In (Jain and Farrokhnia, 1991), it was shown that for texture segmentation, it is usually enough to cluster a portion of this data set, and then assign labels to the rest of the feature vectors according to a nearest neighbor criterion. Therefore, we feed the clustering algorithms with a data set which consists of 10% of the original data set, randomly drawn according to a uniform distribution. Still, this means that the data set to be clustered consists of $N = 6554$ data points. For more details on the Jain and Farrokhnia method to create energy features for texture segmentation, we refer the reader to (Jain and Farrokhnia, 1991).

The textured image to be segmented is shown in Fig. 7 (a). It consists of five different textured regions, with different properties with respect to frequency content and spatial orientation.

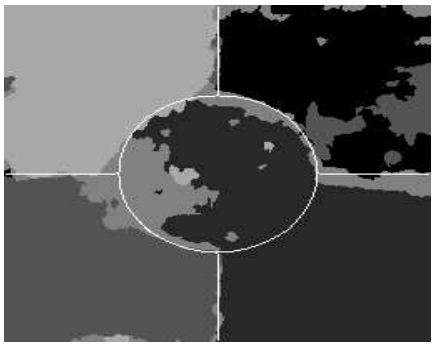
Fig. 7 (b) shows a typical segmentation result obtained by the CS clustering algorithm, operated in annealing mode as described in the beginning of this section. The algorithm



(a) Textured image (256 $\times$ 256)

(b) Typical CS-clustering result. Segmentation errors are located for the most part at texture boundaries.

(c) EMGMM



(d) FKM

Figure 7: Results of texture segmentation experiment.

stops after about $80 - 100$ iterations. The result is very satisfying. The white lines indicate the true cluster boundaries. It can be seen that a few pixels are misclassified, but for the most part a correct segmentation has been performed. The misclassified points are mostly located at texture boundaries, but even in these regions the segmentation is very accurate. Note that the actual pixel locations have not been used as features.

Fig. 7 (c) shows the best result obtained by the EMGMM algorithm. The EMGMM algorithm clearly cannot segment the texture boundaries satisfactorily. Note that the full covariance structure of the Gaussian mixtures is optimized. Optimizing only a diagonal covariance matrix leads to even poorer results. It is also worth mentioning that for this 20 dimensional data set consisting of five clusters, the computational complexity to the EMGMM algorithm is higher than the computational complexity of the CS algorithm. The computational complexity of the CS algorithm does not depend on the dimensionality of the data.

Finally, Fig. 7 (d) shows the best segmentation result obtained by fuzzy K-means. As can be seen, fuzzy K-means performs poorly on this data set. The poor performance of fuzzy K-means indicates that some of the cluster boundaries are non-linear.

5. Conclusions

In this paper, we have developed a practical search algorithm that maximizes an information theoretic clustering measure based on the Cauchy-Schwarz pdf divergence. The appeal of the cost function is that it estimates divergences in continuous probability spaces directly from the data without the need for explicit pdf estimation. The search algorithm possesses very useful properties, namely it has no free parameters, and it is independent of the order of data presentation. Moreover, the free parameter of the cost function, the kernel size in Parzen estimation, was efficiently used as an annealing parameter that avoids local minima during the search. Through stochastic approximation of the gradients used in the CS clustering fixed-point update rule, we have shown that the computational complexity of the algorithm is $O(MN)$, where $M \ll N$. This makes the algorithm capable of clustering relatively large data sets.

Through several experiments, we have shown that the new CS clustering algorithm is capable of producing good clustering results, even when the clusters have irregular and odd-shaped structure, in contrast to many of the popular traditional clustering algorithms. We are presently using a larger set of datasets to characterize better the performance.

Future work will further exploit tools from the density estimation literature in order to incorporate more advanced optimization techniques for the Parzen kernel. Results may be further improved by optimizing the full covariance structure of the Gaussian kernels, instead of using spherical kernels.

It is also important to address the issue of optimizing separate Parzen kernels for each cluster. This may further improve the algorithm in cases where the density of data points vary a lot from cluster to cluster. One possibility is to start out with the same kernel for all clusters, iterate until the rough structure of the clusters has been revealed, for then to optimize separate kernels based on the members assigned to each cluster as the iterative algorithm proceeds.

At present, the annealing scheme that we use is rather heuristic, and therefore still inconsistent in practical applications. In the future, we hope to be able to derive automatic data-driven annealing schemes, which will improve the practical usefulness of the algorithm. Finally, we hope to investigate in more detail the issue of determining the number of clusters in a data set.

Acknowledgments

This work was partially supported by NSF grant ECS-0300340. Robert Jenssen acknowledges the University of Tromsø, for granting a research scholarship in order to visit the University of Florida for the academic year 2002/2003 and for March/April 2004. Deniz Erdogmus and Kenneth E. Hild II were with the Computational NeuroEngineering Laboratory during this work.

Let $\hat{J}_{CS} = \frac{U}{V}$, where

$$\begin{aligned} U &= \frac{1}{2}\sum_{i,j=1}^{N,N}\left(1-\mathbf{m}_i^T\mathbf{m}_j\right)G_{ij,2\sigma^2\mathbf{I}}, \\ V &= \sqrt{\prod_{k=1}^{K} v_k} \text{ and } v_k = \sum_{i,j=1}^{N,N} m_{ik}m_{jk}G_{ij,2\sigma^2\mathbf{I}}. \end{aligned} \tag{26}$$

Hence

$$\frac{\partial \hat{J}_{CS}}{\partial \mathbf{m}_i} = \frac{V\frac{\partial U}{\partial \mathbf{m}_i} - U\frac{\partial V}{\partial \mathbf{m}_i}}{V^2} \tag{27}$$

$$\frac{\partial U}{\partial \mathbf{m}_i} = -\sum_{j=1}^{N}\mathbf{m}_j G_{ij,2\sigma^2\mathbf{I}}, \tag{28}$$

$$\frac{\partial V}{\partial \mathbf{m}_i} = \frac{1}{2}\sum_{k'=1}^{K}\sqrt{\frac{\prod_{k\neq k'}^{K}{}_{k=1} v_k}{v_{k'}}}\frac{\partial v_{k'}}{\partial \mathbf{m}_i}, \tag{29}$$

where $\frac{\partial v_{k'}}{\partial \mathbf{m}_i} = \left[0 \ \ldots \ 2\sum_{j=1}^{N} m_{jk'}G_{ij,2\sigma^2\mathbf{I}} \ \ldots \ 0\right]^T$. Thus, only element number k' of this vector is nonzero.

References

H. M. Abbas and M. M. Fahmy. Neural Networks for Maximum Likelihood Clustering. *Signal Processing*, 36(1):111–126, 1994.

J. C. Bezdek. A Convergence Theorem for the Fuzzy Isodata Clustering Algorithms. *IEEE Transactions on Pattern Analysis and Machine Learning*, 2(1):1–8, 1980.

J. C. Bezdek, M. R. Pal, J. Keller, and R. Krisnapuram. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Kluwer Academic Publishers, Norwell, USA, 1999.

A. Bhattacharyya. On a Measure of Divergence Between Two Statistical Populations Defined by their Probability Distributions. *Bull. Calcutta Math.*, 35:99–109, 1943.

L. Devroye and G. Lugosi. *Combinatorial Methods in Density Estimation*. Springer-Verlag, New York, 2001.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 2nd edition, 2001.

R. P. W. Duin. On the Choice of the Smoothing Parameters for Parzen Estimators of Probability Density Functions. *IEEE Transactions on Computers*, 25(11):1175–1179, 1976.

D. Erdogmus and J. C. Principe. Generalized Information Potential Criterion for Adaptive System Training. *IEEE Transactions on Neural Networks*, 13(5):1035–1044, 2002.

J. H. Friedman and J. W. Tukey. A Projection Pursuit Algorithm for Exploratory Data Analysis. *IEEE Transactions on Computers, Ser. C*, 23:881–889, 1974.

H. Frigui and R. Krishuapuram. A Robust Competitive Clustering Algorithm with Applications in Computer Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):450–465, 1999.

M. Girolami. Mercer Kernel-Based Clustering in Feature Space. *IEEE Transactions on Neural Networks*, 13(3):780–784, 2002.

E. Gokcay and J. Principe. Information Theoretic Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):158–170, 2002.

T. Hofmann and J. M. Buhmann. Pairwise Data Clustering by Deterministic Annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):1–14, 1997.

F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. John Wiley and Sons Ltd, 1999.

A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Englewood Cliffs, 1988.

A. K. Jain and F. Farrokhnia. Unsupervised Texture Segmentation using Gabor Filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

M. C. Jones, J.S. Marron, and S. J. Sheater. A Brief Survey of Bandwidth Selection for Density Estimation. *Journal of the Royal Statistical Society*, 87:227–233, 1996.

D. Kazakos and P. Papantoni-Kazakos. *Detection and Estimation*. Computer Science Press, New York, 1990.

S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by Simulated Annealing. *Science*, pages 671–680, 1983.

S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

J. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, University of California Press, Berkeley, 1967.

O.L Mangasarian and W. H. Wolberg. Cancer Diagnosis via Linear Programming. *SIAM News*, 5:1–18, 1990.

G. J. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, New York, 2000.

N. Morgan and H. Franco. Applications of Neural Networks to Speech Recognition. *IEEE Signal Processing Magazine*, 14(6):46–47, 1997.

R. Murphy and D. Ada. UCI Repository of Machine Learning databases. Technical report, Dept. Comput. Sci. Univ. California, Irvine, 1994.

E. Parzen. On the Estimation of a Probability Density Function and the Mode. *The Annals of Mathematical Statistics*, 32:1065–1076, 1962.

J. Principe, D. Xu, and J. Fisher. Information Theoretic Learning. In *Unsupervised Adaptive Filtering*, volume I, S. Haykin (Ed.), John Wiley & Sons, New York, 2000. Chapter 7.

S. S. Rao. *Engineering Optimization; Theory and Practice*. John Wiley & Sons, 1996.

S. J. Roberts, R. Everson, and I. Rezek. Maximum Certainty Data Partitioning. *Pattern Recognition*, 33:833–839, 2000.

K. Rose, E. Gurewitz, and G. C. Fox. Vector Quantization by Deterministic Annealing. *IEEE Transactions on Information Theory*, 38(4):1249–1257, 1992.

N. N. Schraudolph. Gradient-Based Manipulation of Nonparametric Entropy Estimates. *IEEE Transactions on Neural Networks*, 15(4):828–837, 2004.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, 1986.

M. A. Styblinski and T.-S. Tang. Experiments in Nonconvex Optimization: Stochastic Approximation with Function Smoothing and Simulated Annealing. *Neural Networks*, 3: 467–483, 1990.

S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, San Diego, 1999.

N. Tishby and N. Slonim. Data Clustering by Markovian Relaxation and the Information Bottleneck Method. In *Advances in Neural Information Processing Systems, 13*, pages 640–646, MIT Press, Cambridge, 2001.

S. Watanabe. *Pattern Recognition: Human and Mechanical*. John Wiley & sons, 1985.

Chapter 4

Paper 2:
Spectral Clustering based on Information Theory and Parzen Windowing

Spectral Clustering based on Information Theory and Parzen Windowing

Robert Jenssen[1] ROBERTJ@PHYS.UIT.NO
Deniz Erdogmus[2] DERDOGMUS@IEEE.ORG
Jose C. Principe[3] PRINCIPE@CNEL.UFL.EDU
Torbjørn Eltoft[1] PCTE@PHYS.UIT.NO

1. *Department of Physics*
University of Tromsø, N-9037 Tromsø, Norway

2. *Computer Science and Engineering Department*
Oregon Graduate Institute, OHSU, Portland, OR. 97006, USA

3. *Department of Electrical and Computer Engineering*
University of Florida, Gainesville, FL. 32611, USA

Abstract

We propose a new spectral clustering algorithm that maximizes the information theoretic Cauchy-Schwarz (CS) divergence measure between probability density functions. We show that Parzen windowing is the key component that reveals the CS divergence as a spectral clustering cost function. The Parzen window in combination with a non-negative weighting function also determines the kernel matrix whose eigenstructure in turn defines the kernel induced feature space where the actual clustering takes place. As a by-product of the theory we present, we are able to learn the kernel matrix from the data at hand for data sets of low to moderate dimensionality, making the spectral clustering algorithm automatic with respect to the kernel size. We illustrate the new spectral clustering scheme using several data sets.

Keywords: Information theory, Cauchy-Schwarz divergence, Parzen windowing, Mercer kernel feature space, generalized information cut, spectral clustering.

1. Introduction

In recent years there has been an increasing interest in the research field known as spectral clustering. In spectral clustering, the data partitioning is obtained based on the eigenvalues (the spectrum) and eigenvectors of a suitably chosen kernel matrix. The kernel function defines similarities between data items. In this paper, we restrict the kernel functions to vectorial inputs. Other choices also exist (Shawe-Taylor and Cristianini, 2004).

Spectral clustering dates back at least to Fiedler (1973), who discovered that a graph can be bi-partitioned by thresholding the eigenvector corresponding to the second eigenvalue of the Laplacian matrix. In the last few years, a number of related techniques have

been published (Ding et al., 2001, Shi and Malik, 2000, Perona and Freeman, 1998, Hagen and Kahng, 1991, Pothen et al., 1990, Sarkar and Soundararajan, 2000). These techniques are all based on variants of the graph *cut*, a measure of the cost of partitioning a graph into two pieces. But they use different kernel matrices, and utilize the information contained in the eigenspectrum of the matrices in different manners. Multiway cuts have also been studied (Chang et al., 1994, Meila and Xu, 2004). For other related work, see for example (Weiss, 1999, Kannan et al., 2000, Alpert and Yao, 1995, Azar et al., 2001, Scott and Longuet-Higgins, 1990, Higham and Kibble, 2004, Jenssen et al., 2004). Another direction in spectral clustering was proposed by Ng et al. (2002). In that work they use the eigenvectors of the Laplacian matrix to transform, or map, the input data into a new representation, for then to perform the actual clustering in that space by the C-means technique (MacQueen, 1967). Normalization of the transformed data to unit length is a crucial step in this procedure. Provided the normalization is performed, this algorithm was shown to yield excellent clustering results on several data sets.

Despite the experimental success of spectral clustering, there are some issues which are not completely understood. For example, it is not always clear which criterion that is optimized in the various algorithms, although Meila and Shi (2000) provided an interpretation of the normalized cut algorithm (Shi and Malik, 2000) in terms of Markov random walks. Another issue that is problematic is the construction of a proper kernel matrix. The construction of the kernel matrix should be completely data driven, and should properly reflect similarities between data items. To the current authors knowledge, there exist no widely accepted procedure to create the kernel matrix for spectral clustering [1].

We consider the clustering problem from a seemingly quite different perspective. We want to partition the data set such that an information theoretic (Shannon and Weaver, 1949) separability measure between the clusters is optimized. Examples of information theoretic separability measures include the probability density function (pdf) divergences proposed by Kullback and Leibler (1951), Chernoff (1952) and Bhattacharyya (1943).

In the current exposition, the partitioning is based on the Cauchy-Schwarz (CS) divergence, which was recently proposed by Principe et al. (2000). Assume that $p_1(\mathbf{x})$ is the pdf that describes cluster C_1 and $p_2(\mathbf{x})$ is the pdf of cluster C_2. Then, the CS divergence can be expressed as

$$D(p_1, p_2) = -\log \frac{\langle p_1, p_2 \rangle_u}{\sqrt{\langle p_1, p_1 \rangle_u \langle p_2, p_2 \rangle_u}} \geq 0, \tag{1}$$

where $\langle p_i, p_j \rangle_u \equiv \int p_i(\mathbf{x}) p_j(\mathbf{x}) u(\mathbf{x}) d\mathbf{x},\ i, j = 1, 2$. Here, $u(\mathbf{x})$ is a non-negative weighting function. Principe et al. (2000) estimated this quantity (for $u(\mathbf{x}) \equiv 1$ only) by replacing the densities by their corresponding Parzen window estimators, and used it for pose estimation in SAR imagery and for independent component analysis. The contribution of the current paper is twofold:

1. We show that a Parzen window-based estimator for the CS divergence has a dual expression as a cost function for clustering in a *kernel induced feature space*

$$\hat{D}(p_1, p_2) = -\log \cos \angle(\mathbf{m}_{1_u}, \mathbf{m}_{2_u}), \tag{2}$$

1. In spectral classification, that is, when the user have labeled training data available, there have been proposed methods to learn the kernel matrix (Cristianini et al., 2002a, Bach and Jordan, 2004).

where $\mathbf{m}_{1_u}$ and $\mathbf{m}_{2_u}$ are the mean vectors of cluster C_1 and C_2, respectively, in the kernel feature space. The mapping $\mathbf{\Phi}_u(\mathbf{x}_l)$ of a data point $\mathbf{x}_l$ to the kernel feature space is in practice approximated by the eigenstructure of a corresponding kernel matrix $\mathbf{K}_u$ as (Williams and Seeger, 2001, Bengio et al., 2003)

$$\mathbf{\Phi}_u(\mathbf{x}_l) \approx [\sqrt{\tilde{\lambda}_1} e_{1l}, \ldots, \sqrt{\tilde{\lambda}_N} e_{Nl}]^T, \tag{3}$$

where e_{ml} denotes the lth element of the mth eigenvector of $\mathbf{K}_u$ and $\tilde{\lambda}_m$ is the corresponding eigenvalue, where $\tilde{\lambda}_1 \geq \cdots \geq \tilde{\lambda}_N$. See (Vapnik, 1995, Müller et al., 1997) for an overview of kernel theory.

2. We develop a simple clustering algorithm maximizing Eq. (2) in the kernel induced feature space by assigning cluster memberships to the data. Since the kernel feature space is determined by the eigenvalues, i.e. the *spectrum*, and the eigenvectors of the kernel matrix, our clustering algorithm belongs to the family of *spectral clustering* algorithms.

We show that it is the Parzen window in combination with the weighting function $u(\mathbf{x})$ that determines the Mercer kernel and hence the mapping to the kernel feature space. We propose to use data-driven tools from statistics for Parzen window-size selection. This in turn determines the Mercer kernel-size.

Note that since the logarithm is a monotonic function, maximizing Eq. (2) is equivalent to minimizing $\cos\angle(\mathbf{m}_{1_u}, \mathbf{m}_{2_u})$. This cost function will be denoted the *generalized information cut* (GIC) in the remainder of this paper, because we will show that it is related to the graph theoretic *cut* (Jenssen et al., 2003). Hence, $GIC = \cos\angle(\mathbf{m}_{1_u}, \mathbf{m}_{2_u})$.

This paper is organized as follows. In section 2 we derive the theory behind Eq. (2). In section 3, we propose the new spectral clustering algorithm, which minimizes the generalized information cut in the kernel feature space. Thereafter, in section 4, we discuss a strategy for learning the kernel matrix from the data at hand. In section 5, we perform some clustering experiments. We make our concluding remarks in section 6.

Some of the theoretical results reported in this paper were presented in (Jenssen et al., 2005).

2. From Information Theory to Kernel Feature Spaces

Parzen windowing is a well-known kernel-based density estimation method (Devroye and Lugosi, 2001, Parzen, 1962). Given a set of iid samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ drawn from the true density $f(\mathbf{x})$, the Parzen window estimator for this distribution is defined as

$$\hat{f}(\mathbf{x}) = \frac{1}{N} \sum_{t=1}^{N} W_{\sigma^2}(\mathbf{x}, \mathbf{x}_t). \tag{4}$$

Here, W_{σ^2} is the Parzen window, or kernel, and σ^2 controls the width of the kernel. The Parzen window must integrate to one, and is typically chosen to be a pdf itself, such as the Gaussian kernel. Hence,

$$W_{\sigma^2}(\mathbf{x}, \mathbf{x}_t) = \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} \exp\left\{-\frac{||\mathbf{x} - \mathbf{x}_t||^2}{2\sigma^2}\right\},$$

which we will assume in the rest of this paper. It is easily shown that Eq. (4) is an asymptotically unbiased and consistent estimator provided σ decays to zero at a certain rate as N tends to infinity (Parzen, 1962). In the finite sample case, the kernel size has to be chosen in a trade-off between estimation bias and variance.

2.1 General Kernel Matrix Feature Space

First, let us focus on the numerator of Eq. (1), given by

$$\int p_1(\mathbf{x})p_2(\mathbf{x})u(\mathbf{x})d\mathbf{x} = \int h(\mathbf{x})g(\mathbf{x})d\mathbf{x}, \tag{5}$$

where $h(\mathbf{x}) = u^{\frac{1}{2}}(\mathbf{x})p_1(\mathbf{x})$ and $g(\mathbf{x}) = u^{\frac{1}{2}}(\mathbf{x})p_2(\mathbf{x})$. Assume that each cluster is represented by a set of iid data samples, that is, $C_1 : \{\mathbf{x}_i\}, i = 1, \ldots, N_1$, and $C_2 : \{\mathbf{x}_j\}, j = 1, \ldots, N_2$. Hence, the input data set is the union of C_1 and C_2, that is, $C_1 \cup C_2 = \{\mathbf{x}_t\},\ t = 1, \ldots, N$, for $N = N_1 + N_2$. Note that the index i always points to cluster C_1, while the index j always points to cluster C_2. We propose the following generalized Parzen window-based estimator for the functions $h(\mathbf{x})$ and $g(\mathbf{x})$

$$\hat{h}(\mathbf{x}) = \frac{1}{N_1}\sum_{i=1}^{N_1} u^{\frac{1}{2}}(\mathbf{x}_i)W_{\sigma_1^2}(\mathbf{x}, \mathbf{x}_i), \tag{6}$$

$$\hat{g}(\mathbf{x}) = \frac{1}{N_2}\sum_{j=1}^{N_2} u^{\frac{1}{2}}(\mathbf{x}_j)W_{\sigma_2^2}(\mathbf{x}, \mathbf{x}_j), \tag{7}$$

where σ_1 is the window size associated with the Parzen estimator $\hat{p}_1(\mathbf{x})$ corresponding to cluster C_1, and σ_2 is the window size associated with $\hat{p}_2(\mathbf{x})$ corresponding to cluster C_2. These estimators are asymptotically unbiased and consistent under certain conditions, as shown in Appendix A.

Using these estimators, we have

$$\begin{aligned}
\int \hat{h}(\mathbf{x})\hat{g}(\mathbf{x})d\mathbf{x} &= \int \frac{1}{N_1}\sum_{i=1}^{N_1} u^{\frac{1}{2}}(\mathbf{x}_i)W_{\sigma_1^2}(\mathbf{x}, \mathbf{x}_i)\frac{1}{N_2}\sum_{j=1}^{N_2} u^{\frac{1}{2}}(\mathbf{x}_j)W_{\sigma_2^2}(\mathbf{x}, \mathbf{x}_j)d\mathbf{x} \\
&= \frac{1}{N_1 N_2}\sum_{i,j=1}^{N_1,N_2} u^{\frac{1}{2}}(\mathbf{x}_i)u^{\frac{1}{2}}(\mathbf{x}_j)\int W_{\sigma_1^2}(\mathbf{x}, \mathbf{x}_i)W_{\sigma_2^2}(\mathbf{x}, \mathbf{x}_j)d\mathbf{x} \\
&= \frac{1}{N_1 N_2}\sum_{i,j=1}^{N_1,N_2} u^{\frac{1}{2}}(\mathbf{x}_i)u^{\frac{1}{2}}(\mathbf{x}_j)W_{(\sigma_1^2+\sigma_2^2)}(\mathbf{x}_i, \mathbf{x}_j),
\end{aligned} \tag{8}$$

where in the last step, the convolution theorem for Gaussians has been employed. Similarly, we have

$$\int \hat{h}^2(\mathbf{x})d\mathbf{x} = \frac{1}{N_1^2}\sum_{i,i'=1}^{N_1,N_1} u^{\frac{1}{2}}(\mathbf{x}_i)u^{\frac{1}{2}}(\mathbf{x}_{i'})W_{2\sigma_1^2}(\mathbf{x}_i, \mathbf{x}_{i'}), \tag{9}$$

and, likewise

$$\int \hat{g}^2(\mathbf{x})d\mathbf{x} = \frac{1}{N_2^2}\sum_{j,j'=1}^{N_2,N_2} u^{\frac{1}{2}}(\mathbf{x}_j)u^{\frac{1}{2}}(\mathbf{x}_{j'})W_{2\sigma_2^2}(\mathbf{x}_j,\mathbf{x}_{j'}). \tag{10}$$

Now we define the matrix $\mathbf{K}$, where the elements $K_{ts} = K(\mathbf{x}_t,\mathbf{x}_s),\ t,s = 1,\ldots,N$, are defined as follows. If both $\mathbf{x}_t \in C_1$ and $\mathbf{x}_s \in C_1$, then $K(\mathbf{x}_t,\mathbf{x}_s) = W_{2\sigma_1^2}(\mathbf{x}_t,\mathbf{x}_s)$. If $\mathbf{x}_t \in C_1$ and $\mathbf{x}_s \in C_2$, then $K(\mathbf{x}_t,\mathbf{x}_s) = W_{(\sigma_1^2+\sigma_2^2)}(\mathbf{x}_t,\mathbf{x}_s)$. And if $\mathbf{x}_t \in C_2$ and $\mathbf{x}_s \in C_2$, then $K(\mathbf{x}_t,\mathbf{x}_s) = W_{2\sigma_2^2}(\mathbf{x}_t,\mathbf{x}_s)$. The matrix $\mathbf{K}$ is often called the affinity matrix. We also define the matrix $\mathbf{K}_u$, whose elements are defined as $K_{u_{ts}} = u^{\frac{1}{2}}(\mathbf{x}_t)K(\mathbf{x}_t,\mathbf{x}_s)u^{\frac{1}{2}}(\mathbf{x}_s),\ t,s = 1,\ldots,N$. The approach taken, it to substitute Eq. (8), Eq. (9) and Eq. (10) into the the argument of the logarithm of Eq. (1), to obtain

$$GIC = \frac{\frac{1}{N_1N_2}\sum_{i,j=1}^{N_1,N_2} K_u(\mathbf{x}_i,\mathbf{x}_j)}{\sqrt{\frac{1}{N_1^2}\sum_{i,i'=1}^{N_1,N_1} K_u(\mathbf{x}_i,\mathbf{x}_{i'})\frac{1}{N_2^2}\sum_{j,j'=1}^{N_2,N_2} K_u(\mathbf{x}_j,\mathbf{x}_{j'})}}. \tag{11}$$

We refer to Appendix B for an explanation of the relationship between this expression and the graph *cut*. This relationship was the reason for naming this estimator the *generalized information cut*.

The key point to note, is that $\mathbf{K}$ is a Mercer kernel matrix (Vapnik, 1995), and so is $\mathbf{K}_u$. Hence, $K(\mathbf{x}_t,\mathbf{x}_s) = \langle \mathbf{\Phi}(\mathbf{x}_t), \mathbf{\Phi}(\mathbf{x}_s)\rangle$ where $\mathbf{\Phi}$ is the mapping of the input data to a kernel feature space. Likewise, $K_u(\mathbf{x}_t,\mathbf{x}_s) = \langle \mathbf{\Phi}_u(\mathbf{x}_t), \mathbf{\Phi}_u(\mathbf{x}_s)\rangle$. Hence, we have

$$\begin{aligned} GIC &= \frac{\frac{1}{N_1N_2}\sum_{i,j=1}^{N_1,N_2} \langle \mathbf{\Phi}_u(\mathbf{x}_i), \mathbf{\Phi}_u(\mathbf{x}_j)\rangle}{\sqrt{\frac{1}{N_1^2}\sum_{i,i'=1}^{N_1,N_1} \langle \mathbf{\Phi}_u(\mathbf{x}_i), \mathbf{\Phi}_u(\mathbf{x}_{i'})\rangle \frac{1}{N_2^2}\sum_{j,j'=1}^{N_2,N_2} \langle \mathbf{\Phi}_u(\mathbf{x}_j), \mathbf{\Phi}_u(\mathbf{x}_{j'})\rangle}} \\ &= \frac{\left\langle \frac{1}{N_1}\sum_{i=1}^{N_1} \mathbf{\Phi}_u(\mathbf{x}_i), \frac{1}{N_2}\sum_{j=1}^{N_2} \mathbf{\Phi}_u(\mathbf{x}_j)\right\rangle}{\sqrt{\left\langle \frac{1}{N_1}\sum_{i=1}^{N_1} \mathbf{\Phi}_u(\mathbf{x}_i), \frac{1}{N_1}\sum_{i'=1}^{N_1} \mathbf{\Phi}_u(\mathbf{x}_{i'})\right\rangle \left\langle \frac{1}{N_2}\sum_{j=1}^{N_2} \mathbf{\Phi}_u(\mathbf{x}_j), \frac{1}{N_2}\sum_{j'=1}^{N_2} \mathbf{\Phi}_u(\mathbf{x}_{j'})\right\rangle}} \\ &= \frac{\langle \mathbf{m}_{1_u}, \mathbf{m}_{2_u}\rangle}{\sqrt{\langle \mathbf{m}_{1_u}, \mathbf{m}_{1_u}\rangle \langle \mathbf{m}_{2_u}, \mathbf{m}_{2_u}\rangle}} = \cos\angle(\mathbf{m}_{1_u}, \mathbf{m}_{2_u}). \end{aligned} \tag{12}$$

Thus, the information theoretic divergence divergence measure between pdfs that we started out with turns out to have a dual expression in a kernel feature space determined by the eigenspectrum of the kernel matrix $\mathbf{K}_u$. In that space, the cost function measures *the cosine of the angle between the cluster mean vectors*.

Next, we consider two specific weighting functions.

2.2 Affinity Matrix Feature Space

Let $u(\mathbf{x}) \equiv 1$. Hence, Eq. (6) and Eq. (7) reduce to

$$\hat{h}(\mathbf{x}) = \hat{p}_1(\mathbf{x}) = \frac{1}{N_1}\sum_{i=1}^{N_1} W_{\sigma_1^2}(\mathbf{x},\mathbf{x}_i), \tag{13}$$

$$\hat{g}(\mathbf{x}) = \hat{p}_2(\mathbf{x}) = \frac{1}{N_2}\sum_{j=1}^{N_2} W_{\sigma_2^2}(\mathbf{x},\mathbf{x}_j), \tag{14}$$

the traditional Parzen window estimators for each of the classes. In this case, the resulting estimator is named simply the *information cut* (IC), given by

$$IC = \cos \angle(\mathbf{m}_1, \mathbf{m}_2), \tag{15}$$

where $\mathbf{m}_1$ and $\mathbf{m}_2$ are mean vectors in a kernel feature space determined by the eigenspectrum of the affinity matrix $\mathbf{K}$.

2.3 Laplacian Matrix Feature Space

Now we propose a particular weighting function given by $u(\mathbf{x}) = f^{-1}(\mathbf{x})$, where $f(\mathbf{x})$ is the overall probability density function of the data set. One justification for imposing such a weighting function may be that it provides a connection between the resulting information theoretic cost function and the probability of error. Let us take a closer look at this connection before actually deriving the estimator in this case. Note that $f(\mathbf{x}) = P_1\, p_1(\mathbf{x}) + P_2\, p_2(\mathbf{x})$. Denote the argument of the logarithm of Eq. (1) in this case as P_f, such that

$$P_f = \frac{\int p_1(\mathbf{x}) p_2(\mathbf{x}) f^{-1}(\mathbf{x}) d\mathbf{x}}{\sqrt{\int p_1^2(\mathbf{x}) f^{-1}(\mathbf{x}) d\mathbf{x} \int p_2^2(\mathbf{x}) f^{-1}(\mathbf{x}) d\mathbf{x}}}. \tag{16}$$

Assume that the two clusters are well separated, such that for $\mathbf{x}_i \in C_1$, $f(\mathbf{x}_i) \approx P_1 p(\mathbf{x}_i)$, while for $\mathbf{x}_j \in C_2$, conversely, $f(\mathbf{x}_j) \approx P_2 q(\mathbf{x}_j)$. Specifically, we have

$$\begin{aligned} \int p_1(\mathbf{x}) p_2(\mathbf{x}) f^{-1}(\mathbf{x}) d\mathbf{x} &\approx \int_{C_1} p_1(\mathbf{x}) p_2(\mathbf{x}) f^{-1}(\mathbf{x}) d\mathbf{x} + \int_{C_2} p_1(\mathbf{x}) p_2(\mathbf{x}) f^{-1}(\mathbf{x}) d\mathbf{x} \\ &= \frac{1}{P_1} \int_{C_1} p_2(\mathbf{x}) d\mathbf{x} + \frac{1}{P_2} \int_{C_2} p_1(\mathbf{x}) d\mathbf{x}. \end{aligned} \tag{17}$$

By performing a similar calculation for $\int p_1^2(\mathbf{x}) f^{-1}(\mathbf{x}) d\mathbf{x}$ and $\int p_2^2(\mathbf{x}) f^{-1}(\mathbf{x}) d\mathbf{x}$, we have

$$P_f \approx \sqrt{\frac{P_1}{P_2}} \int_{C_2} p_1(\mathbf{x}) d\mathbf{x} + \sqrt{\frac{P_2}{P_1}} \int_{C_1} p_2(\mathbf{x}) d\mathbf{x}. \tag{18}$$

This expression can be compared to the expression for the probability or error, given by

$$P_e = P_1 \int_{C_2} p_1(\mathbf{x}) d\mathbf{x} + P_2 \int_{C_1} p_2(\mathbf{x}) d\mathbf{x}. \tag{19}$$

Hence, it can be seen that $P_f \approx \frac{P_e}{\sqrt{P_1 P_2}}$.

Next, we derive a Parzen window-based estimator. In this case, let

$$\hat{h}(\mathbf{x}) = \frac{1}{N_1} \sum_{i=1}^{N_1} f^{-\frac{1}{2}}(\mathbf{x}_i) W_{\sigma_1^2}(\mathbf{x}, \mathbf{x}_i), \tag{20}$$

$$\hat{g}(\mathbf{x}) = \frac{1}{N_2} \sum_{j=1}^{N_2} f^{-\frac{1}{2}}(\mathbf{x}_j) W_{\sigma_2^2}(\mathbf{x}, \mathbf{x}_j), . \tag{21}$$

Moreover, estimate any $f(\mathbf{x}_t)$ by the Parzen window estimator

$$\hat{f}(\mathbf{x}_t) = \frac{1}{N}\sum_{s=1}^{N} W_{\sigma^2}(\mathbf{x}_t, \mathbf{x}_s) = d_t. \tag{22}$$

Define the matrix $\mathbf{D} = \text{diag}(d_1, \ldots, d_N)$. Then, the kernel matrix $\mathbf{K}_f$ corresponding to $u(\mathbf{x}) = f^{-1}(\mathbf{x})$ can be expressed as

$$\mathbf{K}_f = \mathbf{D}^{-\frac{1}{2}}\mathbf{K}\mathbf{D}^{-\frac{1}{2}}, \tag{23}$$

where the (t, s) element is given by $K_f(\mathbf{x}_t, \mathbf{x}_s) = \hat{f}^{-\frac{1}{2}}(\mathbf{x}_t)K(\mathbf{x}_t, \mathbf{x}_s)\hat{f}^{-\frac{1}{2}}(\mathbf{x}_s)$. In fact, this matrix is known as the Laplacian matrix [2].

Thus, in this case the resulting cost function measures the cosine of the angle between mean vectors in a kernel feature space determined by the eigenstructure of the Laplacian matrix. For that reason, we name it the *Laplacian information cut* (LIC), given by

$$LIC = \cos\angle(\mathbf{m}_{1_f}, \mathbf{m}_{2_f}). \tag{24}$$

2.4 Extension To Many Clusters

Note also that the above analysis can be easily extended to any number of pdfs/clusters. In the C-cluster case, we define the CS divergence as

$$D(p_1, \ldots, p_C) = \sum_{i=1}^{C-1}\sum_{j>i} \frac{\langle p_i, p_j\rangle_u}{\kappa\sqrt{\langle p_i, p_i\rangle_u \langle p_j, p_j\rangle_u}},$$

where $\kappa = \sum_{c=1}^{C-1} c$.

3. A Spectral Clustering Algorithm

We have shown that the CS divergence is also a clustering cost function in a kernel induced feature space. Assume for a moment that a method for determining a proper kernel matrix exists. In that case, a schematic procedure to cluster a data set can be formulated as follows. Given a set of points $\{\mathbf{x}_t\},\ t = 1, \ldots, N$, in R^d that is to be clustered into C subsets.

1. Construct the kernel matrix $\mathbf{K}_u$.
2. Find $\mathbf{e}_1, \ldots, \mathbf{e}_N$, and $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_N$, the eigenvectors and eigenvalues of $\mathbf{K}_u$.
3. Map $\mathbf{x}_t \rightarrow \mathbf{\Phi}_u(\mathbf{x}_t) \approx \left[\sqrt{\tilde{\lambda}_1}e_{1t}, \ldots, \sqrt{\tilde{\lambda}_N}e_{Nt}\right]^T,\ t = 1, \ldots, N$, by Eq. (3).
4. Cluster the $\mathbf{\Phi}_u(\mathbf{x}_t)$'s into C clusters such that the the generalized information cut is minimized.

2. It is a bit imprecise to refer to $\mathbf{K}_f$ as the Laplacian matrix, as readers familiar with spectral graph theory (Chung, 1997) may recognize, since the definition of the Laplacian matrix is $\mathbf{L} = \mathbf{I} - \mathbf{K}_f$. However, replacing $\mathbf{K}_f$ by $\mathbf{L}$ does not change the eigenvectors, it only changes the eigenvalues from λ_i to $1 - \lambda_i$

5. Finally, assign the original point $\mathbf{x}_t$ to cluster c if and only if $\mathbf{\Phi}_u(\mathbf{x}_t)$ was assigned to cluster c.

Step 4 of this procedure is a crucial issue that we haven't addressed so far. This is the actual clustering, after the data has been mapped to the kernel feature space. *Our aim is to assign labels to the feature space data in order to minimize the generalized information cut, because we know that this corresponds to maximizing the divergence between the cluster pdfs in the input space.*

In order to suggest a suitable algorithm for labeling the data such that the generalized information cut is minimized, we will analyze its properties in more detail. For notational convenience, denote a kernel feature space data point $\mathbf{\Phi}_u(\mathbf{x})$ by $\mathbf{y}$. Also, denote the mean of cluster C_i in the feature space by $\mathbf{m}_i$.

Assume that at a given time a clustering exists yielding the clusters C_1 and C_2 with corresponding mean vectors $\mathbf{m}_1$ and $\mathbf{m}_2$, respectively. The value of the generalized information cut for this clustering is simply denoted GIC. Consider re-assigning some data point $\mathbf{y}$ from C_1 to C_2, hence obtaining the new clusters C_1' and C_2'. It is easily shown that the new mean vectors are given by

$$\begin{aligned} \mathbf{m}_1' &= \frac{1}{N_1 - 1}\left[N_1\mathbf{m}_1 - \mathbf{y}\right], \\ \mathbf{m}_2' &= \frac{1}{N_2 + 1}\left[N_2\mathbf{m}_2 + \mathbf{y}\right]. \end{aligned}$$

Hence, the new value for the cost function, $GIC' = \cos\angle(\mathbf{m}_1', \mathbf{m}_2')$, is given by

$$\begin{aligned} GIC' &= \cos\angle\left(N_1\mathbf{m}_1 - \mathbf{y}, N_2\mathbf{m}_2 + \mathbf{y}\right) \\ &= \frac{(N_1\mathbf{m}_1 - \mathbf{y})^T(N_2\mathbf{m}_2 + \mathbf{y})}{||N_1\mathbf{m}_1 - \mathbf{y}||||N_2\mathbf{m}_2 + \mathbf{y}||}. \end{aligned}$$

Let $\Gamma_1 = ||N_1\mathbf{m}_1 - \mathbf{y}||||N_2\mathbf{m}_2 + \mathbf{y}||$. Hence, we may rewrite GIC' as follows

$$\begin{aligned} GIC' &= \frac{1}{\Gamma_1}\left(N_1N_2\mathbf{m}_1^T\mathbf{m}_2 + N_1\mathbf{m}_1^T\mathbf{y} - N_2\mathbf{m}_2^T\mathbf{y} - ||\mathbf{y}||^2\right) \\ &= \frac{1}{\Gamma_1}\left(\Gamma_2\cos\angle(\mathbf{m}_1, \mathbf{m}_2) + \Gamma_3\cos\angle(\mathbf{m}_1, \mathbf{y}) - \Gamma_4\cos\angle(\mathbf{m}_2, \mathbf{y}) - ||\mathbf{y}||^2\right) \\ &= \frac{\Gamma_2}{\Gamma_1}GIC + \frac{1}{\Gamma_1}\left(\Gamma_3\cos\angle(\mathbf{m}_1, \mathbf{y}) - \Gamma_4\cos\angle(\mathbf{m}_2, \mathbf{y}) - ||\mathbf{y}||^2\right) \end{aligned} \quad (25)$$

where $\Gamma_2 = N_1N_2||\mathbf{m}_1||||\mathbf{m}_2||$, $\Gamma_3 = N_1||\mathbf{m}_1||||\mathbf{y}||$ and $\Gamma_4 = N_2||\mathbf{m}_2||||\mathbf{y}||$.

We are interested in the generalized information cut taking a value as small as possible after reassigning $\mathbf{y}$. Based on Eq. (25), it is clear that for GIC' to be as small as possible, we need to select an $\mathbf{y}$ such that

$$\min_{\mathbf{y}}\left[\cos\angle(\mathbf{m}_1, \mathbf{y}) - \cos\angle(\mathbf{m}_2, \mathbf{y})\right].$$

A natural consequence of this discussion, is to conclude that for a data point $\mathbf{y}$ assigned to cluster C_1, the following should hold: $\cos\angle(\mathbf{m}_1, \mathbf{y}) > \cos\angle(\mathbf{m}_2, \mathbf{y})$.

Extending the above result to the C-cluster case, we suggest the following simple clustering algorithm in the kernel feature space:

- Initialize some mean vectors $\mathbf{m}_c,\ k = 1, \ldots, C$.
- loop until stop
  - for i = 1 to N;
    - Assign $\mathbf{y}_i$ to C_j if $\max_j \cos \angle(\mathbf{m}_j, \mathbf{y}_i)$.

    end for
  - Update the mean vectors based on the new assignments.
  - Evaluate stopping criterion.

  end loop

A suitable stopping criterion may be based on the value of the cost function. For example, the algorithm may stop when the decrease in the cost function is less than some small threshold.

One can readily appreciate the simple structure of this algorithm. It can be seen that the computational complexity of the procedure scales linearly, $O(N)$, with the number N of feature space data samples. Hence, the computational bottleneck of the overall clustering scheme is 1) the construction of the kernel matrix, which is $O(N^2)$, 2) the eigendecomposition procedure, which is $O(N^3)$. The computational complexity of spectral clustering is clearly a problem. Notice that a promising method to remedy this drawback has been proposed by Fowlkes et al. (2004), using the Nyström method.

Our main focus in this exposition is not computational complexity, but rather to introduce an information theoretic spectral clustering algorithm. Still, we note that it is crucial to be able to properly select the kernel sizes based on the data at hand. Of course, one could guess the kernel sizes, construct the kernel matrix, eigendecompose and perform the clustering for a range of kernel sizes, and then determine the best clustering based on the value of the generalized information cut. In practice, such an approach would be computationally exhaustive. We will return to kernel size selection in section 4.

We also note that the proposed spectral clustering algorithm is a member of the family of hard clustering algorithms based on function optimization (Theodoridis and Koutroumbas, 1999). The assignments of the data patterns corresponds to maximizing the following cost function

$$J(u) = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij} \cos \angle(\mathbf{m}_j, \mathbf{y}_i),$$

where $u_{ij} = 1$ if $\mathbf{y}_i$ is assigned to $\mathbf{m}_j$, and zero otherwise.

It is easily seen by visual inspection that if $J(u)$ is maximized, then the GIC is minimized. Consider for example Fig. 1. The data points assigned to C_1 is marked by '$\circ$' and the data points assigned to C_2 by '$\bullet$'. The mean vectors are marked by the squares. For this clustering, $J(u)$ obviously takes its maximum value. Also, the GIC takes its minimum value. If, for example, any data point is reassigned from C_1 to C_2, $J(u)$ will decrease and GIC will increase.

The proposed clustering algorithm in the kernel feature space will be demonstrated in section 5. First, we discuss some issues which pertain to the dimensionality of the feature space data samples, to the initialization of the feature space mean vectors and to the estimation of the number of clusters, C, present in the data.

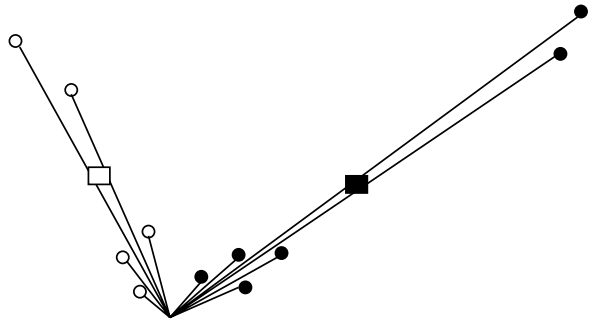


Figure 1: For this clustering, $J(u)$ takes its maximum value and the GIC takes its minimum value. For any reassignment of data patterns from one cluster to the other, $J(u)$ will decrease and the GIC will increase. The squares indicates the mean vectors for the two clusters, respectively.

3.1 Dimensionality Reduction in Feature Space

By Eq. (3), it can be seen that the kernel feature space data is N-dimensional. Hence, for very large N it may be problematic to execute the clustering algorithm. By performing an "ideal" case analysis, that is, by considering the clusters to be "infinitely" far apart, it can easily be shown that the kernel feature space data is C-dimensional in the ideal case, since there are only C non-zero eigenvalues. In the real case, if there are C distinct clusters one may therefore suspect that most of the information is associated with the C largest eigenvalues.

Assume therefore that the input data is mapped into a feature space which is truncated to only C-dimensions. That is, $\mathbf{\Phi}_u(\mathbf{x}_t) \approx \left[\sqrt{\tilde{\lambda}_1}e_{1t}, \ldots, \sqrt{\tilde{\lambda}_C}e_{Ct}\right]^T$, $t = 1, \ldots, N$, $\tilde{\lambda}_1 \geq \cdots \geq \tilde{\lambda}_C$. Brand and Huang (2003) showed that such a dimensionality reduction in the feature space corresponds to approximating $\mathbf{K}_u$ with a matrix $\hat{\mathbf{K}}_u$ which is the best rank-C approximation to $\mathbf{K}_u$ with respect to the Frobenius norm. See Appendix C for more on this discussion.

We use this approximation because it lowers the computational burden by reducing the data dimensionality from N to C. Also, in our experience it does not reduce the clustering capability of the algorithm.

3.2 Initializing the Mean Vectors

Consider again Fig. 1. We can observe that if two criteria are satisfied with respect to the structure of the feature space data set, then the algorithm may converge very fast to the correct solution. These criteria are: 1) The data points belonging to each cluster are distributed more or less along separate lines radially from the origin, where the angles between all pairs of lines are sufficiently large. 2) The mean vectors are initialized such that each mean vector is close to one of these lines, and each line has an initial mean vector nearby.

In the "ideal" case (Appendix C), we observe that the C-dimensional mean vectors are proportional to $\mathbf{m}_1 = [\pm 1, 0, \ldots, 0]^T$, $\mathbf{m}_2 = [0, \pm 1, \ldots, 0]^T$, and so on. If the mean vectors are initialized along each of these orthogonal directions, the data will be correctly clustered by our algorithm in one iteration. In the real case, we choose to initialize the mean vectors based on this knowledge. Note that the first coordinate, or channel, of the feature space data depends on the first eigenvalue/eigenvector. Hence, if the mean value of the first eigenvector is positive, we initialize $\mathbf{m}_1 = [1, 0, \ldots, 0]^T$. Otherwise $\mathbf{m}_1 = [-1, 0, \ldots, 0]^T$. Likewise, we examine the mean of the second eigenvector and initialize the corresponding mean vector based on the sign of that number, and so on. We have found this initialization to be very robust.

3.3 Estimating the Number of Clusters

Girolami (2002) proposed a method for estimating the number of clusters in a data set based on Renyi's quadratic entropy, $H(\mathbf{x}) = -\log \int f^2(\mathbf{x}) d\mathbf{x}$, of the data and the eigendecomposition of the kernel matrix. The CS divergence is closely related to the Renyi entropy (Principe et al., 2000). Using the traditional Parzen window method, Girolami showed that $V(\mathbf{x}) = \int f^2(\mathbf{x}) d\mathbf{x}$ can be estimated by

$$V(\mathbf{x}) \approx \frac{1}{N^2} \sum_{i=1}^{N} \tilde{\lambda}_i \{\mathbf{1}_N^T \mathbf{e}_i\}^2, \tag{26}$$

where $\mathbf{1}_N$ is a N-dimensional all ones vector and $\tilde{\lambda}_i / \mathbf{e}_i$ is the ith eigenvalue/eigenvector of the kernel matrix $\mathbf{K}$. Girolami showed for several datasets that if there are C distinct clustered regions within the N data samples then there will be C dominant terms $\lambda_i \{\mathbf{1}^T \mathbf{e}_i\}^2$ in the summation Eq. (26).

4. Learning the Kernel Matrix

In our framework, the kernel matrix $\mathbf{K}_u$ depends explicitly on the Parzen estimator. Hence, if we know the optimal kernel sizes σ, $\sigma_1, \sigma_2, \ldots, \sigma_C$, and the weighting components are given, we can construct the kernel matrix $\mathbf{K}_u$. However, one problem we immediately face, is that we don't know which data points belong to which cluster (since this is exactly what we are trying to determine), so that it is impossible to obtain a separate kernel size σ_i for each cluster C_i. Given an input data set, the best we can do is to estimate the optimal (in some sense) kernel size σ based on the whole data set. This kernel size corresponds to the traditional Parzen estimator $\hat{f}(\mathbf{x})$ for the overall pdf of the data. Consequently, we select $\sigma_i = \sigma$, for all the clusters, even though this choice may not be optimal for the individual clusters.

Parzen kernel size selection has been thoroughly studied in the statistics literature (Silverman, 1986, Scott, 1992, Wand and Jones, 1995). The optimal kernel size is usually selected so as to minimize the mean integrated squared error (MISE) between $\hat{f}(\mathbf{x})$ and the target density $f(\mathbf{x})$. It is easily shown (Silverman, 1986, Scott, 1992, Wand and Jones, 1995) that the MISE decomposes into a bias term and a variance term. For a fixed sample size, the bias term is minimized by minimizing the kernel size, while the variance term is min-

imized by maximizing the kernel size. This is the inherent bias-variance trade-off in the Parzen window technique.

A well-known leave-one-out search for the kernel size which minimizes the MISE is known as unbiased least squares cross-validation (LSCV) (Wand and Jones, 1995). The estimator based on the resulting σ_{LSCV} has been analyzed and found to have a large variance, that is, this kernel size is often too small (Scott, 1992).

Another straight-forward approach is to find the kernel size which minimizes the asymptotic MISE (AMISE). By assuming that the underlying density is Gaussian, an expression for the optimal kernel size is given by (Silverman, 1986) $\sigma_{\text{AMISE}} = \sigma_X \left[\frac{4}{(2d+1)N}\right]^{\frac{1}{d+4}}$, where $\sigma_X^2 = d^{-1}\sum_i \mathbf{\Sigma}_{X_{ii}}$, and $\mathbf{\Sigma}_{X_{ii}}$ are the diagonal elements of the sample covariance matrix. The main appeal of this approach is that is is very easy to use. The obvious drawback is that it assumes that the underlying density is unimodal and Gaussian. Therefore, σ_{AMISE} is often too large, corresponding to a too smooth density estimate.

Based on the discussion above, one possible approach for kernel size selection is to combine the two aforementioned approaches, using

$$\sigma = \frac{1}{2}\left(\sigma_{\text{AMISE}} + \sigma_{\text{LSCV}}\right). \tag{27}$$

In this paper we will use Eq. (27) for selecting the kernel size used in Parzen window density estimation. Since this kernel size also specifies the corresponding kernel matrix, *the kernel matrix is learned from the data at hand*. This makes our spectral clustering algorithm automatic with respect to kernel size selection.

Due to the "curse-of-dimensionality" (Scott, 1992, Silverman, 1986) the practical use of the kernel size selectors discussed above is unfortunately limited only to data sets of low to moderate dimensionality. The curse-of-dimensionality refers to the fact that the usual bias-variance trade-off cannot be accomplished very well in higher dimensions without very large samples.

However, as we will show later, we are able to obtain promising clustering results also for higher-dimensional data sets by manually selecting a kernel size which is larger than the MISE-optimal. This will be illustrated in section 5. Friedman (1997) discussed this issue in the context of classification and showed that the bias and variance have quite different roles when the density estimation is used in a classification rule. For classification, low variance is much more important than low bias, hence favoring a relatively large kernel size. This makes sense also in clustering, since we are seeking a density estimate, biased or not, which is smooth enough to preserve the relative structure of the clusters.

5. Experimental Analysis

In this section we perform some spectral clustering experiments. We have created four artificial data sets. These data sets are all bivariate, making it easy to visualize the clustering results. The data sets resemble some of the challenging data sets used in (Ng et al., 2002). We will compare with the Ng et al. (2002) spectral clustering algorithm, which represents the state-of-the-art in the literature. We also include some experiments using the real Iris, Pendigit and Wisconsin Breast cancer data sets, extracted from the UCI repository (Murphy and Ada, 1994).

For the lower dimensional data sets, we operate the new spectral clustering algorithm in a fully automatic mode with respect to the kernel size. The kernel size, σ, is given by Eq. (27). Based on the discussion in section 2, we note that the effective kernel size used in the kernel matrix construction is $\tilde{\sigma} = \sqrt{2}\sigma$. For some of the real data sets, we will analyze the clustering results for a range of kernel sizes. Note that given a kernel size, there is no random component in our algorithm. The mean vector initialization is deterministic. Therefore, *there is no variation in the clustering results by running the algorithm several times, using the same kernel size in each case.* We choose to stop the clustering algorithm if the decrease in the generalized information cut cost function between two consecutive steps is less than 10^{-4}. We perform the clustering in kernel feature spaces determined by the eigenspectrums of the affinity matrix and the Laplacian matrix, respectively. The eigendecomposition of the matrices is performed using MATLAB.

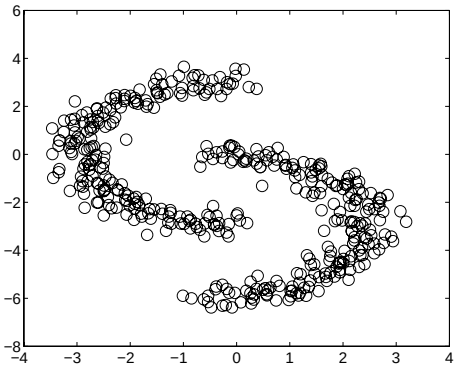
When comparing with the Ng et al. (2002) algorithm, we use the same effective kernel size, $\tilde{\sigma} = \sqrt{2}\sigma$, where σ is determined by Eq. (27). To further make the algorithms comparable, we initialize the mean vectors associated with this algorithm in the same manner as proposed in subsection 3.2.

5.1 Bivariate Data Sets

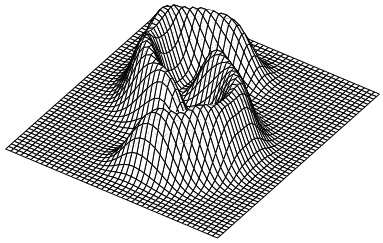
The first data set we examine is formed as two "half moons", as shown in Fig. 2 (a). It can be seen that this is a highly non-Gaussian data set, where both clusters are relatively dense. It contains a total of $N = 419$ data patterns. The leftmost cluster contains $N_1 = 209$ patterns, while the rightmost cluster contains $N_2 = 210$ patterns. We estimate the kernel size using Eq. (27), yielding $\sigma = 0.55$. The corresponding Parzen pdf estimate for this data set is shown in Fig. 2 (b). The density estimate clearly shows the main structure of the data.

First, we use the proposed method to estimate the number of clusters. Consider Fig. 2 (c). It shows the first 10 $\lambda_i\{\mathbf{1}^T\mathbf{e}_i\}^2$ terms in the summation Eq. (26) using the automatically selected kernel size. It can be seen that there are two terms which dominate significantly more than the others, indicating $C = 2$ clusters present in the data. In our experience, this example is representative for a wide variety of data sets where the clusters have more or less the same density of data points. In that case the method reliably estimates the number of clusters. If the density of data points in the clusters varies significantly, the method is less robust.

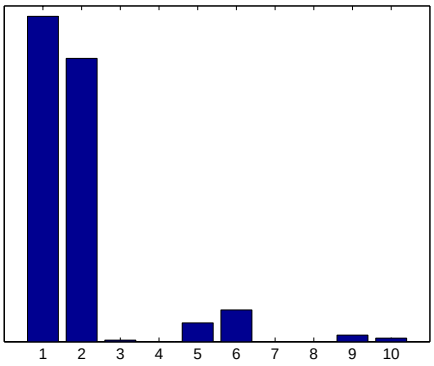
Let us next take a closer look at the kernel feature space data that is given by the eigenspectrum of the affinity matrix, $\mathbf{K}$. Note that the affinity matrix is constructed using the automatically selected kernel size, $\sigma = 0.55$. Since there are $C = 2$ clusters, the feature space data is two-dimensional, making it easy to visualize. Fig. 3 (a) shows a scatter plot of the resulting feature space data. The affinity matrix feature space data is for the most part distributed along two lines radially from the origin, almost perpendicular to each other. These two lines are indicated in the figure. By running the spectral clustering algorithm we have proposed on this feature space data, the two clusters indicated by the 'squares' and 'stars', respectively, are obtained. This result is in fact obtained after *only one iteration*, indicating that the mean vectors are properly initialized. Of course, the interesting part is how this feature space clustering relates to the input space data. This is shown in Fig. 3



(a) Input data set



(b) Parzen pdf estimate

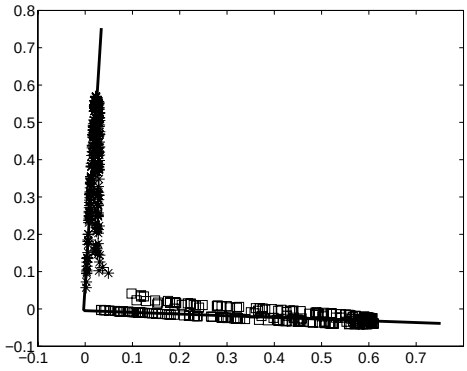


(c) Estimating the number of clusters

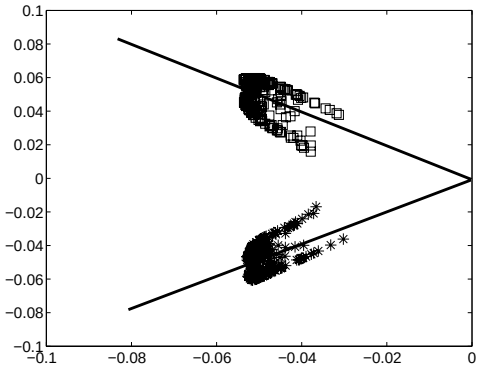
Figure 2: (a) "Half moon" data set. Both clusters are dense. (b) Corresponding Parzen window estimate of the overall data density. The kernel size is automatically selected, yielding $\sigma = 0.55$. (c) Estimating the number of clusters based on Renyi's entropy. The 10 largest terms in Eq. (26) are shown. There are two dominant terms, indicating $C = 2$ clusters present in the data set.

(c). The clustering result is in total agreement with the clustering a human would obtain by visual inspection of the input data set.

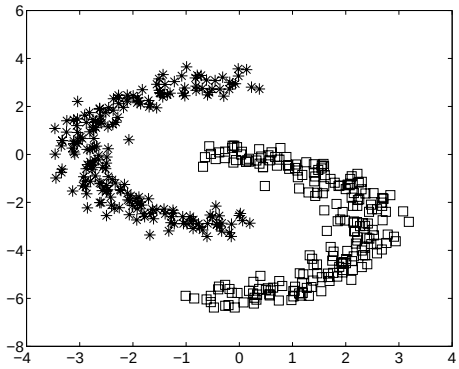
When mapping the input data based on the Laplacian kernel matrix, the feature space data shown in Fig. 3 (b) is obtained. Also in this case, the data is distributed for the most part in two different angular directions in feature space. After two iterations, the data is clustered such that the data patterns marked with 'squares' belong to one cluster, and the



(a) Affinity matrix feature space



(b) Laplacian matrix feature space

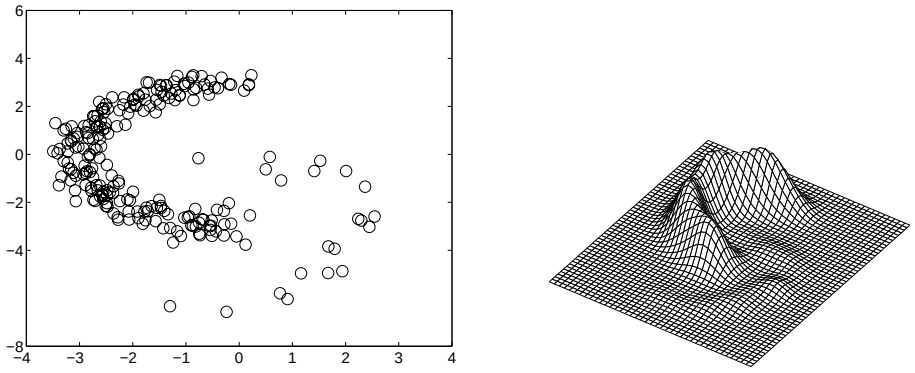


(c) Clustering result

Figure 3: (a), (b) Kernel feature space data sets for a mapping given by the eigenstructure of the affinity matrix and the Laplacian matrix, respectively. (c) Clustering result obtained in both cases.

'stars' to the other cluster. In fact, this clustering also corresponds to the input space data clustering shown in Fig. 3 (c), which is the correct result.

The results we have presented are for automatically generated kernel matrices, i.e. we as users have not specified the kernel size to be used. We have seen that our clustering algorithm produces a perfect clustering result both for the feature space data being generated from the affinity matrix and the Laplacian matrix. As an experiment, we investigate how robust the algorithm is with respect to the kernel size. In the affinity matrix case, we find that the algorithm produce a perfect clustering result for $0.4 < \sigma < 0.65$. In the Laplacian matrix case, the corresponding range of kernel sizes is $0.1 < \sigma < 0.65$. Hence, in this



(a) Input data set

(b) Parzen pdf estimate

Figure 4: (a) "Half moon" data set. One cluster is dense, the other is sparse. (b) Corresponding Parzen window estimate of the overall data density. The kernel size is automatically selected, yielding $\sigma = 0.51$.

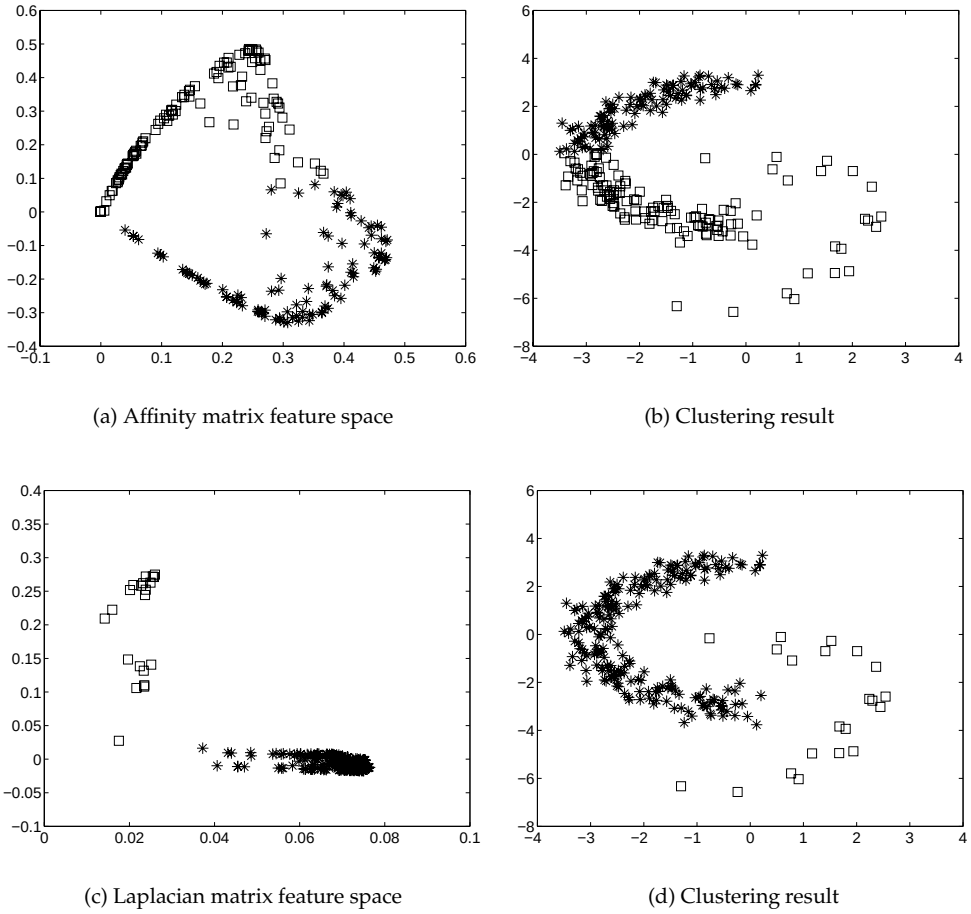
case the clustering turns out to be more robust for the Laplacian mapping. For kernel sizes outside these ranges, the kernel feature space data starts to overlap, such that a perfect clustering result can not be obtained.

When comparing with the Ng et al. (2002) approach on this data set, we observe that this algorithm performs very similar to our algorithm based on the Laplacian matrix. Hence, we have not shown these results.

The second data set we examine is also formed as two "half moons", as shown in Fig. 4 (a). However, in this case there is one dense cluster and one very sparse cluster. The data set contains a total of $N = 230$ data patterns. The dense cluster contains $N_1 = 209$ patterns, while the sparse cluster contains only $N_2 = 21$ patterns. The automatically selected kernel size yields $\sigma = 0.51$. The corresponding Parzen pdf estimate for this data set is shown in Fig. 4 (b). It can be seen that it is only the structure of the dense cluster which is clearly visible.

For this data set, the Girolami (2002) method for estimating the number of clusters indicates that there is only one cluster. This is not very surprising, considering Fig. 4 (b). Therefore, in the following experiments, we have specified $C = 2$ as an input parameter to our clustering algorithm.

The spectrally transformed data set based on the eigendecomposition of the affinity matrix for $\sigma = 0.51$ is shown in Fig. 5 (a). Also, the clusters revealed by minimizing the information cut in the feature space is shown, marked by 'squares' and 'stars'. Actually, considering the structure of the feature space data, the clustering obtained in that space looks reasonable with respect to minimizing the cost function. However, as shown in Fig. 5 (b), the corresponding clusters in the input space is not in agreement with the data structure a human can observe.



(a) Affinity matrix feature space

(b) Clustering result

(c) Laplacian matrix feature space

(d) Clustering result

Figure 5: (a) Affinity matrix kernel feature space, and (b) corresponding clustering result. (c) Laplacian matrix kernel feature space, and (d) corresponding clustering result.

Clearly, there is nothing obviously "wrong" with the actual clustering in the feature space. The data mapping just does not allow for the data to be clustered correctly, with respect to the structure of the data in the input space. In fact, even by manually selecting the kernel size over a wide range, the clustering scheme based on the affinity matrix does not produce the sought-for result.

The spectrally transformed data set based on the eigendecomposition of the Laplacian matrix for $\sigma = 0.51$ is shown in Fig. 5 (c). The feature space clustering is also shown. In contrast to the affinity matrix case, this feature space clustering corresponds to a perfect input space clustering, as shown in Fig. 5 (d). Moreover, the same perfect clustering result

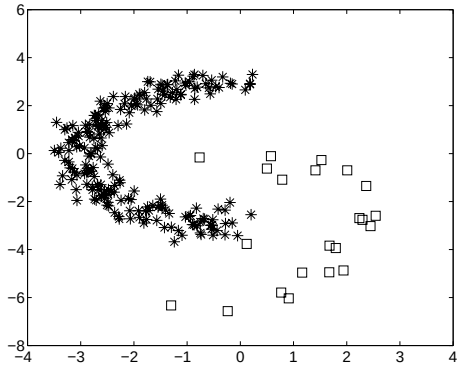


Figure 6: Clustering result using the Ng et al. (2002) approach.

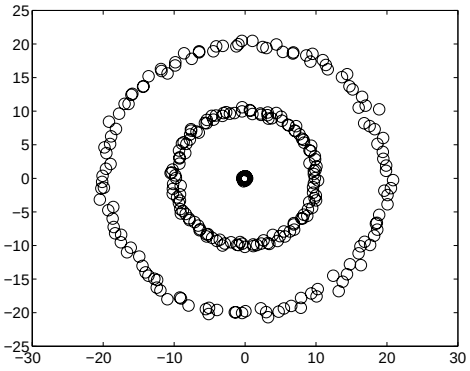
is obtained for a relatively wide range of kernel sizes. The completely different results for the affinity matrix and the Laplacian matrix cases, respectively, are obviously caused mostly by the data transformations being very different.

The clustering result using the Ng et al. (2002) algorithm for $\sigma = 0.51$ is shown in Fig. 6. It can bee seen that the result is very similar to the result obtained using the proposed algorithm based the Laplacian matrix. Only one data point has been clustered differently.

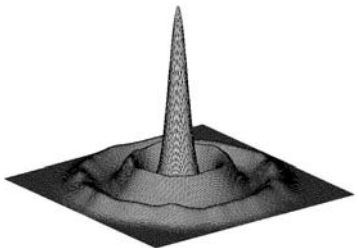
The third data set we examine consists of three rings, or circles. The circles have different radius. The data set is shown in Fig. 7 (a). There are a total of $N = 315$ data patterns. The inner-most circle contains $N_1 = 63$ data points. The radius is so small that it looks almost like a point-cluster, hence it is very dense. The middle circle and the outer circle both contain $N_2 = N_3 = 126$ data patterns. Thus, the outer circle is the most sparse cluster. The kernel size is determined to be $\sigma = 1.9$ by Eq. (27). The corresponding overall Parzen pdf estimate is shown in Fig. 7 (b).

Fig. 7 (c) shows the contribution to the Renyi entropy from the terms in Eq. (26). In fact, the method only detects two clusters. This shows that this technique is of limited value when the clusters differ substantially in density. Still, it is interesting to apply our clustering algorithm to the data set for $C = 2$ clusters. Fig. 7 (d) shows the clustering result that is based on the eigendecomposition of the affinity matrix. It can be seen that the algorithm combines the two outer-most rings into one cluster, and the inner-most ring as the other cluster. Fig. 8 (a) shows the clustering result that is based on the eigendecomposition of the Laplacian matrix. Here, the two inner-most rings have been combined into one cluster, whereas the outer-most ring constitute the other cluster. By specifying $C = 3$ as an input parameter to the clustering algorithm based on the Laplacian matrix, the result shown in Fig. 8 (b) is obtained. This is the correct result. The same result is obtained using the Ng et al. (2002) approach. When clustering based on the affinity matrix for $C = 3$, the correct result is not obtained.

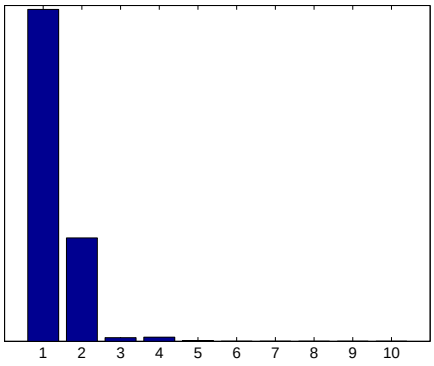
The fourth data set is shown in Fig. 9 (a). A very similar data set was only bipartitioned in (Ng et al., 2002), using a search procedure for the kernel size based on the value of the C-means cost function in feature space. The clustering obtained by our algorithm is indicated by the different symbols. The same clustering is obtained based on both the affinity matrix



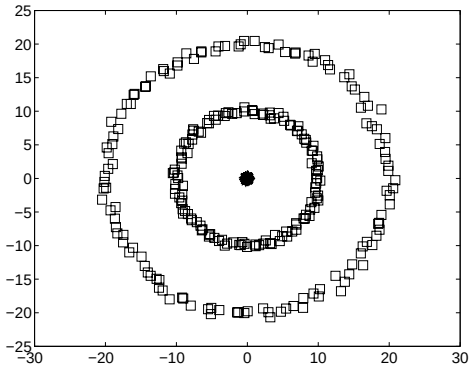
(a) Input data set



(b) Parzen pdf estimate



(c) There are two dominant entropy terms, indicating $C = 2$.



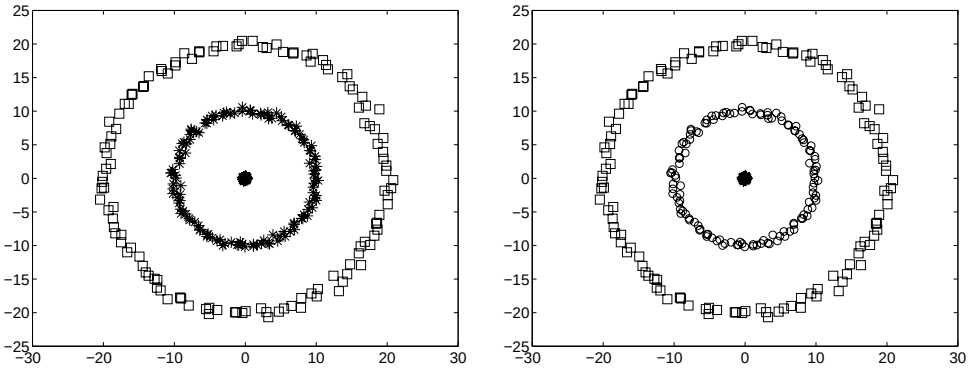
(d) Clustering result based on the affinity matrix for $C = 2$.

Figure 7: (a) "Rings" data set. (b) Corresponding Parzen window estimate of the overall data density. The kernel size is automatically selected, yielding $\sigma = 1.9$.

and the Laplacian matrix for a kernel size which is automatically determined to be $\sigma = 0.36$. Fig. 9 (b) shows that for this data set there are clear indications of four clusters present. Note that the clusters have roughly equal density. In fact, our implementation of the Ng et al. (2002) algorithm also gives the correct result.

5.2 Iris Data Set

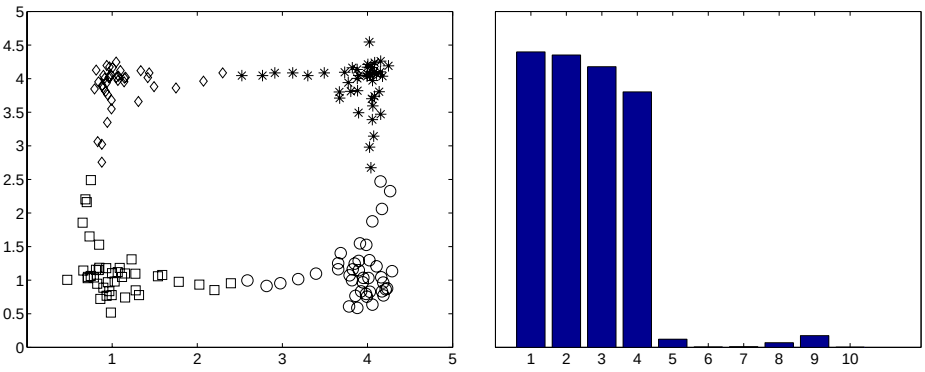
In this experiment we cluster the Iris data set. We include this data set since it is very well-known. The "true" labeling for the Iris data set is available (not to the algorithm), so we know that it contains three classes of 50 patterns each, where each class refers to a type



(a) Clustering result based on the Laplacian matrix for $C = 2$.

(b) Clustering result based on the Laplacian matrix for $C = 3$.

Figure 8: Clustering results for the "rings" data set.



(a) Input data set and resulting clustering

(b) There are four dominant entropy terms, indicating $C = 4$.

Figure 9: (a) Input data set and resulting clustering. (b) Entropy terms.

of iris plant. It is characterized by four numeric attributes. The Iris data set is known to be difficult to cluster, since two of the classes overlap, and the boundaries between these two classes are non-linear. In fact, these two classes overlap to such a degree that based on the structure of the data alone one would probably conclude that there are only two clusters present, and not three. The Girolami method for estimating the number of clusters also indicate only two clusters present, for a kernel size which is automatically determined to be $\sigma = 0.32$. Since we in this case already know that there are in fact three clusters, we feed $C = 3$ as an input parameter to the spectral clustering algorithm.

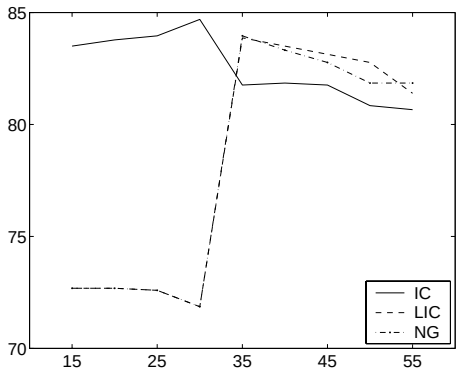


Figure 10: Success rate of clustering the Pendigits data set (in percent) versus kernel size (horizontal axis).

The clustering result obtained based on the affinity matrix yields 10 errors, corresponding to 93.3% correct clustering, as compared to the true labeling. This is in fact a reasonable satisfying result, taking into consideration that there are no user specified parameters apart from having specified the number of clusters. The clustering based on the Laplacian matrix results in 14 errors, which is also the case for the Ng et al. (2002) approach (90.7% correct clustering). We tried to vary the kernel size manually to determine whether the clustering results would improve by choosing a smaller or larger kernel size. We only observed small differences, but no improvement. For a kernel size deviating significantly from $\sigma = 0.32$, the results gets increasingly worse.

5.3 Pendigits Data Set

This data set was created for pen-based handwritten digit recognition (Alimoglu, 1996). The data set is 16-dimensional. All attributes are integers in the range $[0, 100]$. From the test data, we extract the data vectors corresponding to the digits 0, 1 and 2. These classes consist of 363, 364 and 364 data patterns, respectively. The kernel size is automatically determined to be $\sigma = 19.7$. We specify $C = 3$ as an input parameter to the algorithm. In this case, the result based on the affinity matrix correspond to 83.7% correct clustering. We compare to the true labels, which we have available to assess performance. The result based on the Laplacian matrix and the Ng et al. (2002) algorithm are identical, obtaining 72.6% correct clustering. Hence, we may be tempted to conclude that the clustering based on the affinity matrix is the best. But for such a higher-dimensional data set, we should be cautious, since the kernel size selection procedure becomes more uncertain. Fig. 10 shows a plot of the clustering result for the three methods as a function of the kernel size. Here, 'IC' denotes *information cut* (affinity matrix), 'LIC' denotes *Laplacian information cut* and 'NG' denotes the Ng et al. (2002) algorithm. The IC method performs better than 80% over the whole range, but decreases as the kernel size increases. The best result is 84.7% correct clustering. The other two methods perform almost identically to each other. For a kernel

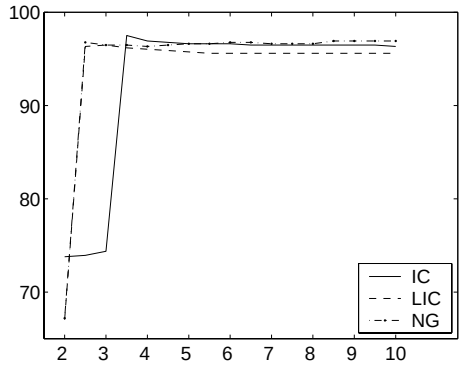


Figure 11: Success rate of clustering the Wisconsin breast-cancer data set (in percent) versus kernel size (horizontal axis).

size about $\sigma = 30$, the performance of both methods significantly improves. For a 'large' kernel size, the LIC and NG methods perform better than the IC method.

5.4 Wisconsin Breast Cancer Data Set

The Wisconsin breast-cancer (WBC) data set (Mangasarian and Wolberg, 1990) consists of two classes of tumors, namely *benign* and *malignant*. There are totally 683 data points, where 444 correspond to the benign class and 239 to the malignant class. WBC is a nine-dimensional dataset with features related to clump thickness, uniformity of cell size and shape, etc.

The kernel size is automatically selected to be $\sigma = 1.57$. However, none of the methods perform well for this kernel size. In order to obtain reasonable results, we need to manually select a kernel size $\sigma > 2$. Fig. 11 shows a plot of the success rate for the methods as a function of kernel size. If can be seen that the LIC and NG methods perform quite similar. For $\sigma \geq 2.5$ these methods have a success rate greater than 95%. The IC method perform well for $\sigma \geq 3.5$. The best result is 97.5% correct clustering. For kernel sizes greater than about $\sigma = 10$, the performance decreases.

We note that Cristianini et al. (2002b) optimized the *cut* cost for clustering this data set. Results were presented using a Gaussian kernel size $\tilde{\sigma} = 6$ to construct the affinity matrix. A success rate of 80.3% was reported. Note that in our framework a kernel size $\sigma \approx 4$ corresponds to an effective kernel $\tilde{\sigma} = \sqrt{2} \times 4 \approx 6$.

6. Conclusions

We have

- Derived a novel spectral clustering algorithm that maximizes the information theoretic Cauchy-Schwarz pdf divergence cost function, and we have demonstrated its simplicity and promising performance.

- Proposed a method for learning the kernel matrix for data sets of low to moderate dimensionality.
- Shown that the proposed spectral clustering algorithm may also work well for higher dimensional data sets, although we at present have no reliable automatic procedure for selecting the kernel size in high dimensions.
- Incorporated a method for estimating the number of clusters. This method fits well into the information theoretic framework of the current exposition.

Based on the above mentioned points, we emphasize that the spectral clustering algorithm we have proposed optimizes a well defined information theoretic cost function, and may be operated in a fully automatic mode. But as we have seen, there are weak points. The limitations posed by the "curse-of-dimensionality" makes proper kernel size selection in higher dimensional data spaces difficult. Since the Parzen window estimators used in our algorithm are parts of a clustering system, we may have to shift focus from kernel size selection for optimal density estimation, to kernel size selection for optimal clustering. As mentioned, Friedman (1997) have shown that for classification it is more important to obtain a Parzen density estimator having low variance ("large" kernel size) than low bias, a property which may be investigated further with respect to automatic kernel size selection also in high-dimensional data spaces. Also, the estimation of the number of clusters is not good enough at present, and should be investigated further.

For the artificial data sets, we observed that the clustering results based on the Laplacian matrix were sometimes superior to those obtained using the affinity matrix, especially if the density of data points varied significantly from cluster to cluster. Consider again the estimator of for example $h(\mathbf{x})$ associated with the Laplacian information cut

$$\hat{h}(\mathbf{x}) = \frac{1}{N_1} \sum_{i=1}^{N_1} f^{-\frac{1}{2}}(\mathbf{x}_i) W_{\sigma_1^2}(\mathbf{x}, \mathbf{x}_i). \tag{28}$$

Recall that the information cut estimator is obtained using

$$\hat{h}(\mathbf{x}) = \hat{p}_1(\mathbf{x}) = \frac{1}{N_1} \sum_{i=1}^{N_1} W_{\sigma_1^2}(\mathbf{x}, \mathbf{x}_i). \tag{29}$$

We note that if the $f(\mathbf{x}_i)$'s are small, the weighting on the corresponding kernels will be larger in the Laplacian information cut estimator than in the information cut. Hence, it seems that clusters corresponding to a region where the overall probability density have small values will somehow be emphasized more in the Laplacian information cut estimator than in the information cut.

For the real data sets we evaluated, the experiments showed that the clustering results based on the affinity matrix and the Laplacian matrix were quite similar. The reason for this result may be that the clusters do not vary significantly in density in these cases. The relationship between the affinity matrix and the Laplacian matrix clearly needs to be addressed in future work.

We have seen that the clustering results we have obtained using the new spectral clustering algorithm are comparable to the results obtained using *our implementation* of the

heuristically motivated spectral clustering algorithm proposed by Ng et al. (2002). This may to some degree be expected since both methods cluster the data in a space determined by the eigenvectors of the Laplacian matrix, even though the data mapping is not exactly similar in the two cases, since we in our case incorporate the eigenvalues in the mapping. In practice, since we truncate the kernel feature space, the values of the retained eigenvalues do not differ that much. Recall that the top eigenvalue of the Laplacian matrix as defined in this paper is one. It should also be realized that C-means clustering of *normalized data* really amounts to clustering based on an angular measure. In a sense, therefore, the Ng et al. (2002) approach approximates our information theoretic spectral clustering algorithm.

Acknowledgments

This work was partially supported by NSF grant ECS-0300340. Deniz Erdogmus was with the Computational NeuroEngineering Laboratory during this work. Robert Jenssen would like to express gratitude to the Department of Mathematical Sciences at the University of Tromsø, for granting a research scholarship for the academic year 2002/2003, and a two-month research scholarship in the spring of 2004, for visiting the Computational NeuroEngineering Laboratory at the University of Florida.

Appendix A: Asymptotic Properties of Generalized Parzen Estimator

In this appendix we give conditions for the generalized Parzen kernel estimator of $h(\mathbf{x}) = v(\mathbf{x})f(\mathbf{x})$ to be asymptotically unbiased and consistent. Recall that

$$\hat{h}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} v(\mathbf{x}_i)W_{\sigma^2}(\mathbf{x},\mathbf{x}_i), \tag{30}$$

where $\mathbf{x}_1,\ldots,\mathbf{x}_N$ are iid samples drawn from $f(\mathbf{x})$.

This estimator is asymptotically unbiased, which can be shown as follows

$$\begin{aligned} E_f\left\{\hat{h}(\mathbf{x})\right\} &= E_f\left\{\frac{1}{N}\sum_{i=1}^{N} v(\mathbf{x}_i)W_{\sigma^2}(\mathbf{x},\mathbf{x}_i)\right\} \\ &= E_f\left\{v(\mathbf{z})W_{\sigma^2}(\mathbf{x},\mathbf{z})\right\} \\ &= \int v(\mathbf{z})f(\mathbf{z})W_{\sigma^2}(\mathbf{x},\mathbf{z})d\mathbf{z} \\ &= [v(\mathbf{x})f(\mathbf{x})] * W_{\sigma^2}(\mathbf{x}), \end{aligned} \tag{31}$$

where $E_f(\cdot)$ denotes expectation with respect to the density $f(\mathbf{x})$. In the limit as $N \to \infty$ and $\sigma(N) \to 0$, we have

$$\lim_{\substack{N\to\infty \\ \sigma(N)\to 0}} [v(\mathbf{x})f(\mathbf{x})] * W_{\sigma^2}(\mathbf{x}) = v(\mathbf{x})f(\mathbf{x}). \tag{32}$$

We have that

$$\begin{aligned}
\mathrm{Var}_f\left\{\hat{h}(\mathbf{x})\right\} &= \frac{1}{N}\mathrm{Var}_f\left\{v(\mathbf{z})W_{\sigma^2}(\mathbf{x},\mathbf{z})\right\} \\
&= \frac{1}{N}\left(E_f\left\{[v(\mathbf{z})W_{\sigma^2}(\mathbf{x},\mathbf{z})]^2\right\} - \left(E_f\left\{v(\mathbf{z})W_{\sigma^2}(\mathbf{x},\mathbf{z})\right\}\right)^2\right) \\
&\leq \frac{1}{N}E_f\left\{[v(\mathbf{z})W_{\sigma^2}(\mathbf{x},\mathbf{z})]^2\right\} \\
&= \frac{1}{N}\int f(\mathbf{z})v^2(\mathbf{z})W^2_{\sigma^2}(\mathbf{x},\mathbf{z})d\mathbf{z}. \qquad (33)
\end{aligned}$$

Let $W_{\sigma^2}(\zeta) \equiv \frac{1}{\sigma}\tilde{W}(\frac{\zeta}{\sigma})$, where $\sup_\zeta \tilde{W}(\frac{\zeta}{\sigma}) = c$. Hence

$$\begin{aligned}
\mathrm{Var}_f\left\{\hat{h}(\mathbf{x})\right\} &\leq \frac{1}{N}\int f(\mathbf{z})v^2(\mathbf{z})\frac{1}{\sigma^2}\tilde{W}^2\left(\frac{\mathbf{x},\mathbf{z}}{\sigma}\right)d\mathbf{z} \\
&\leq \frac{c^2}{N\sigma^2}\int f(\mathbf{z})v^2(\mathbf{z})d\mathbf{z} \\
&= \frac{c^2}{N\sigma^2}E_f\left\{v^2(\mathbf{x})\right\}. \qquad (34)
\end{aligned}$$

Thus, the sufficient conditions for $\hat{h}(\mathbf{x})$ to be consistent (and asymptotically unbiased) are:

$$E_f\left\{v^2(\mathbf{x})\right\} < \infty, \quad \lim_{N\to\infty}\sigma(N) = 0, \quad \lim_{N\to\infty}N\sigma^2(N) = \infty. \qquad (35)$$

Appendix B: Connection to the Graph Cut

A graph consists of the node set C, with a symmetric similarity weight s_{ij} between nodes i and j. A graph can be partitioned into two disjoint sets C_1 and C_2 simply by removing edges between the two parts. The degree of similarity between these two pieces can be computed as a total weight of the edges that have been removed. This quantity is called the *cut* (Shi and Malik, 2000). Assume that there are N_1 nodes in subgraph C_1 and N_2 nodes in subgraph C_2. Then the *cut* can be expressed as

$$\mathrm{CUT}(C_1, C_2) = \sum_{i,j=1}^{N_1,N_2} s_{ij}. \qquad (36)$$

The cut has been proposed by Wu and Leahy (1993) as a cost function for clustering, where the aim is to minimize it. It has also been used as a cost function in spectral clustering (Cristianini et al., 2002b).

However, as was also noted in (Wu and Leahy, 1993), the minimum cut criteria favors a skewed cut, isolating one or a few nodes in C_1, and the rest in C_2. To compensate for this fact, a number of rather heuristically motivated improvements to the cut-cost have been proposed. All of these methods aim to minimize the similarity between sub graphs, i.e. the cut, while the similarity within each sub graph is maximized. Specifically, we mention the normalized cut (Shi and Malik, 2000), the min-max cut (Ding et al., 2001), the typical cut (Gdalyahu et al., 2001) and the BCut (Scanlon and Deo, 1999).

In the graph theoretic literature, the similarity weight between nodes i and j is commonly given by

$$s_{ij} = \exp\left\{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma_s^2}\right\}, \tag{37}$$

where $\mathbf{x}_i$ is the spatial coordinate of node i, $\mathbf{x}_j$ is the spatial coordinate of node j and σ_s is a scale parameter.

We repeat the expression for the generalized information cut estimator, as

$$GIC = \frac{\sum_{i,j=1}^{N_1,N_2} K_u(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{\sum_{i,i'=1}^{N_1,N_1} K_u(\mathbf{x}_i, \mathbf{x}_{i'}) \sum_{j,j'=1}^{N_2,N_2} K_u(\mathbf{x}_j, \mathbf{x}_{j'})}}. \tag{38}$$

We will now discuss why this estimator was called the generalized information cut.

First, let $u(\mathbf{x}) = 1,\ \forall \mathbf{x}$. In that case, recall that

$$K_u(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j) = W_{2\sigma^2}(\mathbf{x}_i, \mathbf{x}_j). \tag{39}$$

For simplicity, let $\sqrt{2}\sigma = \tilde{\sigma}$. Then

$$W_{\tilde{\sigma}^2} = \frac{1}{(2\pi\tilde{\sigma}^2)^{\frac{d}{2}}} \exp\left\{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\tilde{\sigma}^2}\right\}. \tag{40}$$

In fact, the constant $\frac{1}{(2\pi\tilde{\sigma}^2)^{\frac{d}{2}}}$ can be disregarded, since it cancels out in (38).

We note that the numerator of (38), for $u(\mathbf{x}) = 1$, is exactly equal to (36) when similarity weights are defined by (37). That is, the numerator of (38) is identical to the *cut* known from graph theory. However, from our information theoretic starting point, we have revealed that the *cut* should be normalized by the quantity in the denominator of (38). Consider for example $\sum_{i,i'=1}^{N_1,N_1} K(\mathbf{x}_i, \mathbf{x}_{i'})$. In a graph theoretic framework this double summation adds the similarity weights between all nodes in one of the subgraphs, say subgraph C_1. This quantity is often called the volume of that subgraph.

In (Jenssen et al., 2003), this connection was first noted. Since our starting point is an information theoretic distance measure, which turns out to be closely connected to the graph theoretic cut, the resulting estimator was called the *Information Cut* (IC). In a graph theoretic notation, it can be expressed as

$$IC(C_1, C_2) = \frac{CUT(C_1, C_2)}{\sqrt{VOL(C_1)VOL(C_2)}}. \tag{41}$$

In this exposition, $u(\mathbf{x})$ can be any positive weighting function. Recall that $K_u(\mathbf{x}_i, \mathbf{x}_j) = u^{\frac{1}{2}}(\mathbf{x}_i)K(\mathbf{x}_i, \mathbf{x}_j)u^{\frac{1}{2}}(\mathbf{x}_j)$. Here, $u^{\frac{1}{2}}(\mathbf{x}_i)$ can be considered as a weighting of the importance of node i. Similarly, $u^{\frac{1}{2}}(\mathbf{x}_j)$ weights the importance of node j. The function $u(\mathbf{x})$ can be considered to convey some apriori information about the nodes in the graph, or equivalently about the input data patterns. For example, outliers in the data set may be given a small importance. That way, even if two outliers, $\mathbf{x}_i$ and $\mathbf{x}_j$ are close, such that $K(\mathbf{x}_i, \mathbf{x}_j)$ is large, $K_u(\mathbf{x}_i, \mathbf{x}_j)$ may be small. Based on these general properties of $u(\mathbf{x})$, we call the resulting estimator (38) the *generalized information cut*.

Appendix C: "Ideal" Case Analysis

In order to get some intuition about the spectral data mapping, it is instructive to perform an "ideal" case analysis. That is, we consider the mapping in the case that the clusters are "infinitely" far apart. We focus on the two-cluster case. First, we consider the spectral mapping given by the eigenstructure of the affinity matrix, $\mathbf{K}$.

Moving the clusters "infinitely" far apart, corresponds to zeroing all the elements K_{ij} corresponding to points $\mathbf{x}_i$ and $\mathbf{x}_j$ in different clusters. Elements K_{ij} for points in the same cluster will be equal to some constant. Without loss of generality let $K_{ij} = 1$ in this case. Hence, the affinity matrix is a block diagonal matrix

$$\mathbf{K} = \begin{bmatrix} \underline{\underline{\mathbf{1}}}^{N_1 \times N_1} & \underline{\underline{\mathbf{0}}}^{N_1 \times N_2} \\ \underline{\underline{\mathbf{0}}}^{N_2 \times N_1} & \underline{\underline{\mathbf{1}}}^{N_2 \times N_2} \end{bmatrix},$$

where $\underline{\underline{\mathbf{1}}}$ is the all ones matrix, $\underline{\underline{\mathbf{0}}}$ is the all zero matrix, and the size of each block is indicated. Furthermore, since $\underline{\underline{\mathbf{1}}}^{N_1 \times N_1}$ is a rank-one matrix, it has only one eigenvector, which we denote $\mathbf{e}^{(1)}$. It is straightforward to show that $\mathbf{e}^{(1)} = \pm\frac{1}{\sqrt{N_1}}\mathbf{1}_{N_1}$, where $\mathbf{1}_{N_1}$ is a N_1-dimensional column vector consisting of all ones. The corresponding eigenvalue is $\tilde{\lambda}^{(1)} = N_1$. Similarly, for $\underline{\underline{\mathbf{1}}}^{N_2 \times N_2}$, we have $\mathbf{e}^{(2)} = \pm\frac{1}{\sqrt{N_2}}\mathbf{1}_{N_2}$ and $\tilde{\lambda}^{(2)} = N_2$. Consequently, $\mathbf{E}$ is given by

$$\mathbf{E} = \begin{bmatrix} \pm\frac{1}{\sqrt{N_1}}\mathbf{1}_{N_1} & \mathbf{0}_{N_1} \\ \mathbf{0}_{N_2} & \pm\frac{1}{\sqrt{N_2}}\mathbf{1}_{N_2} \end{bmatrix},$$

and $\mathbf{\Lambda} = \text{diag}(N_1, N_2)$, where $\mathbf{\Lambda}$ is the diagonal matrix with the eigenvalues of $\mathbf{K}$ in decreasing order on its diagonal. Hence, it has been assumed that $N_1 \geq N_2$. The end result of this discussion is that when each data point is mapped into feature space by (3), we obtain

$$\mathbf{\Phi}(\mathbf{x}_i) = \begin{Bmatrix} [\pm 1, 0]^T & \text{if } \mathbf{x}_i \in C_1 \\ [0, \pm 1]^T & \text{if } \mathbf{x}_i \in C_2 \end{Bmatrix}.$$

A similar analysis for the Laplacian data matrix $\mathbf{K}_f$ yields the mapping

$$\mathbf{\Phi}_f(\mathbf{x}_i) = \begin{Bmatrix} \left[\pm\frac{1}{N_1}, 0\right]^T & \text{if } \mathbf{x}_i \in C_1 \\ \left[0, \pm\frac{1}{N_2}\right]^T & \text{if } \mathbf{x}_i \in C_2 \end{Bmatrix}.$$

Thus, in the ideal case, the data mapping based on both these kernel matrices create point clusters which are orthogonal to each other in the kernel feature space. In the input space, such a situation corresponds to the pdfs not overlapping at all.

Note that in the ideal two-cluster case, the data is mapped into a two-dimensional space. In the ideal C-cluster case, the data would be mapped into a C-dimensional space. Hence, in the ideal case, the data is mapped into a feature space of a dimensionality which is independent on the dimensionality of the input data, only dependent on the number of clusters in the data.

In the real world, the pdfs of the clusters will in general have some overlap, and the kernel matrices will have more than C non-zero eigenvalues, even if there are C distinct

clusters present. Therefore, the dimensionality of the kernel feature space data will in general be N-dimensional.

Assume that we map the input data to a feature space which is truncated to only C-dimensions. That is, $\mathbf{x}_t \rightarrow \mathbf{\Phi}_\alpha(\mathbf{x}_t) = \left[\sqrt{\tilde{\lambda}_1}e_{1t}, \ldots, \sqrt{\tilde{\lambda}_C}e_{Ct}\right]^T$, $t = 1, \ldots, N$, $\tilde{\lambda}_1 > \cdots > \tilde{\lambda}_C$. Define

$$\underline{\underline{\mathbf{\Phi}_\alpha}}^{N\times N} = \left[\mathbf{\Phi}_\alpha(\mathbf{x}_1), \ldots, \mathbf{\Phi}_\alpha(\mathbf{x}_N)\right].$$

By (3), we have $\underline{\underline{\mathbf{\Phi}_\alpha}} = \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{E}^T$. Suppose that we choose to represent $\underline{\underline{\mathbf{\Phi}_\alpha}}$ by a truncated version $\underline{\underline{\hat{\mathbf{\Phi}}_\alpha}}$, where $\underline{\underline{\hat{\mathbf{\Phi}}_\alpha}}$ is the top C rows of $\underline{\underline{\mathbf{\Phi}_\alpha}}$. This corresponds to the truncated data mapping above. A well-known property of such a truncated eigendecomposition is that

$$\hat{\mathbf{K}}_\alpha = \underline{\underline{\hat{\mathbf{\Phi}}_\alpha}}^T \underline{\underline{\hat{\mathbf{\Phi}}_\alpha}},$$

is the best rank-C approximation to $\mathbf{K}_\alpha$ with respect to the Frobenius norm (Brand and Huang, 2003). Equivalently, it is the most energy-preserving projection of rank C. Now, the C-dimensional mean vectors are initialized in the exact same manner as in the "ideal" case.

This approximation implies loss of information in the mapping to the kernel feature space, but has the advantage that it drastically reduces the dimensionality of the data in the feature space from N to C.

References

F. Alimoglu. Combining Multiple Classifiers for Pen-Based Handwritten Digit Recognition. Master's thesis, Institute of Graduate Studies in Science and Engineering, Bogazici University, Turkey, 1996.

C. Alpert and S. Yao. Spectral Partitioning: The More Eigenvectors the Better. In *Proceedings of ACM/IEEE Design Automation Conference*, San Francisco, USA, June 12-16, 1995.

Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia. Spectral Analysis of Data. In *Proceedings of ACM Symposium on Theory of Computing*, pages 619–626, Heraklion, Greece, June 6-8, 2001.

F. Bach and M. I. Jordan. Learning Spectral Clustering. In *Advances in Neural Information Processing Systems, 16*, MIP Press, Cambridge, 2004.

Y. Bengio, P. Vincent, and J.-F. Paiement. Spectral Clustering and Kernel PCA are Learning Eigenfunctions. Technical report, Département d'informatique et recherche opérationelle, université de Montréal, Montréal, Canada, 2003.

A. Bhattacharyya. On a Measure of Divergence Between Two Statistical Populations Defined by their Probability Distributions. *Bull. Calcutta Math.*, 35:99–109, 1943.

M. Brand and K. Huang. A Unifying Theorem for Spectral Embedding and Clustering. In *Proceedings of International Workshop on Artificial Intelligence and Statistics*, Key West, USA, January 3-6, 2003.

P. Chang, D. Schlag, and J. Zien. Spectral K-Way Ratio-Cut Partitioning and Clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(9):1088–1096, 1994.

H. Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on a Sum of Observations. *The Annals of Mathematical Statistics*, 23:493–507, 1952.

F. R. K. Chung. *Spectral Graph Theory*. American Mathematial Society, 1997.

N. Cristianini, J. Shawe-Taylor, A. Eliesseeff, and J. Kandola. On Kernel-Target Alignment. In *Advances in Neural Information Processing Systems, 14*, pages 367–373, MIT Press, Cambridge, 2002a.

N. Cristianini, J. Shawe-Taylor, and J. Kandola. Spectral Kernel Methods for Clustering. In *Advances in Neural Information Processing Systems, 14*, pages 649–655, MIT Press, Cambridge, 2002b.

L. Devroye and G. Lugosi. *Combinatorial Methods in Density Estimation*. Springer-Verlag, New York, 2001.

C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A Min-max Cut Algorithm for Graph Partitioning and Data Clustering. In *Proceedings of IEEE International Conference on Data Mining*, pages 107–114, San Jose, USA, November 29 - December 2, 2001.

M. Fiedler. Algebraic Connectivity in Graphs. *Czechoslovak Mathematics Journal*, 23:298–305, 1973.

C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral Grouping using the Nyström Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:298–305, 2004.

J. H. Friedman. On Bias, Variance, 0/1 Loss, and the Curse-Of-Dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997.

Y. Gdalyahu, D. Weinshall, and M. Werman. Self-Organization in Vision: Stochastic Clustering for Image Segmentation, Perceptual Grouping, and Image Database Organization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1053–1074, 2001.

M. Girolami. Mercer Kernel-Based Clustering in Feature Space. *IEEE Transactions on Neural Networks*, 13(3):780–784, 2002.

L. Hagen and A. B. Kahng. Fast Spectral Methods for Ratio Cut Partitioning and Clustering. In *Proceedings of IEEE International Conference on Computer-Aided Design*, pages 10–13, Santa Clara, USA, November 11-14, 1991.

D. J. Higham and M. Kibble. A Unified View of Spectral Clustering. Technical Report 2, University of Strathclyde, Department of Mathematics, January 2004.

R. Jenssen, T. Eltoft, and J. C. Principe. Information Theoretic Spectral Clustering. In *Proceedings of International Joint Conference on Neural Networks*, pages 111–116, Budapest, Hungary, July 25-29, 2004.

R. Jenssen, D. Erdogmus, J. C. Principe, and T. Eltoft. The Laplacian PDF Distance: A Cost Function for Clustering in a Kernel Feature Space. In *Advances in Neural Information Processing Systems 17 (to appear)*, MIT Press, Cambridge, 2005.

R. Jenssen, J. C. Principe, and T. Eltoft. Information Cut and Information Forces for Clustering. In *Proceedings of IEEE International Workshop on Neural Networks for Signal Processing*, pages 459–468, Toulouse, France, September 17-19, 2003.

R. Kannan, S. Vempala, and A. Vetta. On Clusterings: Good, Bad and Spectral. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, pages 367–377, Redondo Beach, USA, November 12-14, 2000.

S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

J. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, University of California Press, Berkeley, 1967.

O.L Mangasarian and W. H. Wolberg. Cancer Diagnosis via Linear Programming. *SIAM News*, 5:1–18, 1990.

M. Meila and J. Shi. Learning Segmentation by Random Walks. In *Advances in Neural Information Processing Systems, 12*, pages 873–897, MIT Press, Cambridge, 2000.

M. Meila and L. Xu. Multiway Cuts and Spectral Clustering. Technical Report 442, University of Washington, Department of Statistics, January 2004.

K. R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. N. Vapnik. Predicting Time Series with Support Vector Machines. In *Proceedings of International Conference on Artificial Neural Networks - Lecture Notes in Computer Science, Springer-Verlag*, volume 1327, pages 999–1004, Berlin, 1997.

R. Murphy and D. Ada. UCI Repository of Machine Learning databases. Technical report, Dept. Comput. Sci. Univ. California, Irvine, 1994.

A. Y. Ng, M. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems, 14*, pages 849–856, MIT Press, Cambridge, 2002.

E. Parzen. On the Estimation of a Probability Density Function and the Mode. *The Annals of Mathematical Statistics*, 32:1065–1076, 1962.

P. Perona and W. T. Freeman. A Factorization Approach to Grouping. In *Proceedings of European Conference on Computer Vision*, pages 655–670, Freiburg im Breisgay, Germany, June 2-6, 1998.

A. Pothen, H. D. Simon, and K. P. Liou. Partitioning Sparse Matrices with Eigenvectors of Graphs. *SIAM Journal of Matrix Analysis and Applications*, 11(3):430–452, 1990.

J. Principe, D. Xu, and J. Fisher. Information Theoretic Learning. In *Unsupervised Adaptive Filtering*, volume I, S. Haykin (Ed.), John Wiley & Sons, New York, 2000. Chapter 7.

S. Sarkar and P. Soundararajan. Supervised Learning of Large Perceptual Organization: Graph Spectral Partitioning and Learning Automata. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):504–525, 2000.

J. Scanlon and N. Deo. Graph-Theoretic Algorithms for Image Segmentation. In *IEEE International Symposium on Circuits and Systems*, pages VI141–144, Orlando, Florida, 1999.

D. W. Scott. *Multivariate Density Estimation*. John Wiley & Sons, New York, 1992.

G. Scott and H. Longuet-Higgins. Feature Grouping by Relocalisation of Eigenvectors of the Proximity Matrix. In *Proceedings of British Machine Vision Conference*, pages 103–108, Oxford, UK, September 24-27, 1990.

C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, 1949.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, 1986.

S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, San Diego, 1999.

V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

M. P. Wand and M. C. Jones. *Kernel Smooting*. Chapman and Hall, London, 1995.

Y. Weiss. Segmentation Using Eigenvectors: A Unifying View. In *Proceedings of IEEE International Conference on Computer Vision*, pages 975–982, Corfu, Greece, September 20-25, 1999.

C. Williams and M. Seeger. Using the Nyström Method to Speed Up Kernel Machines. In *Advances in Neural Information Processing Systems, 13*, pages 682–688, MIT Press, Cambridge, 2001.

Z. Wu and R. Leahy. An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Applications to Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993.

Chapter 5

Paper 3:
Some Equivalences Between Kernel Methods and Information Theoretic Methods

Some Equivalences Between Kernel Methods and Information Theoretic Methods

Robert Jenssen[1] ROBERTJ@PHYS.UIT.NO
Deniz Erdogmus[2] DERDOGMUS@IEEE.ORG
Jose C. Principe[3] PRINCIPE@CNEL.UFL.EDU
Torbjørn Eltoft[1] PCTE@PHYS.UIT.NO

1. *Department of Physics*
University of Tromsø, N-9037 Tromsø, Norway

2. *Computer Science and Engineering Department*
Oregon Graduate Institute, OHSU, Portland, OR. 97006, USA

3. *Department of Electrical and Computer Engineering*
University of Florida, Gainesville, FL. 32611, USA

Abstract

In this paper we discuss some equivalences between two recently introduced statistical learning schemes, namely Mercer kernel methods and information theoretic methods. We show that Parzen window-based estimators for some information theoretic cost functions are also cost functions in a corresponding Mercer kernel space. The Mercer kernel is directly related to the Parzen window. Furthermore, we propose a new classification rule based on an information theoretic criterion, and show that this corresponds to a linear classifier in the kernel space. We then introduce a weighted Parzen window density estimator, and formulate the SVM classifier in an information theoretic perspective.

Keywords: Information theoretic learning, Mercer kernel feature space, Parzen windowing, support vector machine, regularization.

1. Introduction

During the last decade, research on Mercer kernel-based learning algorithms has flourished (Shawe-Taylor and Cristianini, 2004, Müller et al., 2001, Perez-Cruz and Bousquet, 2004, Schölkopf and Smola, 2002). These algorithms include for example the support vector machine (SVM) (Cortez and Vapnik, 1995, Vapnik, 1995, Cristianini and Shawe-Taylor, 2000, Burges, 1998, Hastie et al., 2004), kernel principal component analysis (KPCA) (Schölkopf et al., 1998) and kernel Fisher discriminant analysis (KFDA) (Mika et al., 1999, Roth and Steinhage, 2000). The common property of these methods is that they are linear in nature, as they are being explicitly expressed in terms of inner-products. However, they may be applied to *non-linear* problems using the so-called "kernel-trick". The kernel trick refers to the technique of computing inner-products in a potentially infinite-dimensional

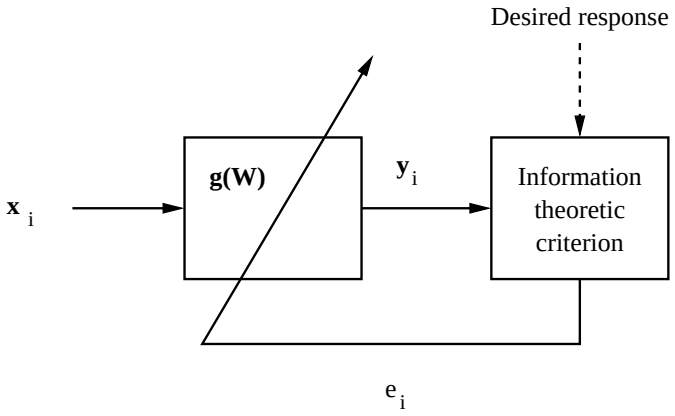


Figure 1: Information theoretic learning setup.

kernel feature space, using so-called Mercer kernels. Mercer kernel-based methods have been applied successfully in several applications, e.g. pattern and object recognition (LeCun et al., 1995), time series prediction (Müller et al., 1997) and DNA and protein analysis (Zien et al., 2000), to name a few.

The Mercer kernel-based methods rely on the assumption that the data becomes easier to handle after the transformation to the Mercer kernel feature space. In the case of the SVM, the assumption is that the data classes become linearly separable, and therefore a separating hyperplane can be created. In practice, one can not know for certain that this assumption holds. In fact, one has to hope that the user chooses a kernel which turns out to properly separate the data.

Independent of the research on Mercer kernel-based learning algorithms another very powerful machine learning scheme has emerged. This is coined *information theoretic learning* (Principe et al., 2000a,b). In information theoretic learning, the starting point is a data set that globally conveys information about a real-world event. The goal is to capture the information in the parameters of a learning machine, using some *information theoretic performance criterion*. A typical setup for information theoretic learning is shown in Figure 1. The system output is given by $\mathbf{y}_i = \mathbf{g}(\mathbf{W})\mathbf{x}_i$, where $\mathbf{x}_i$ is the data pattern presented to the system at iteration i. The function $\mathbf{g}(\mathbf{W})$ represents a possibly non-linear data transformation, which depends on a parameter matrix $\mathbf{W}$. At each iteration, the criterion is evaluated and a correction term e_i generated, which is fed back to the system to guide the adjustment of the system parameters. The system may receive external input in the form of a desired response, in which case the system operates in a supervised learning mode.

The mean squared error criterion (MSE) has traditionally been the workhorse of adaptive systems training (Haykin, 2000). However, the great advantage of information theoretic criteria is that they are able to capture higher order statistical information in the data, as opposed to the MSE, which is a second order statistical criterion. This property is important, since recently many machine learning problems have been encountered where the MSE criterion is insufficient. Such problems include blind source separation and inde-

pendent component analysis, blind equalization and deconvolution, subspace projections, dimensionality reduction, feature extraction, classification and clustering.

Information theoretic criteria are expressed as integrals over functions of probability densities. One possible approach to evaluate such criteria for an observed data set is to replace the densities by density estimators. Using parametric density estimators may be problematic, because they often require numerical integration procedures to be developed. Parzen windowing (Parzen, 1962, Devroye, 1989, Silverman, 1986, Scott, 1992, Wand and Jones, 1995) has been proposed as an appropriate density estimation technique, since this method makes no parametric assumptions. Viola et al. (1995) proposed to approximate Shannon-based measures using sample means, integrated with Parzen windowing (Viola and Wells, 1997). Principe et al. (2000a) went a step further, by introducing a series of information theoretic quantities which can be estimated without the sample mean approximation (Xu, 1999, Principe et al., 2000b). This is important, since the sample mean approximation may not hold very well for small sample sizes. The proposed measures were all based on the generalizations of the Shannon entropy derived by Renyi (1976b,a), and include Renyi's quadratic entropy, the Cauchy-Schwarz (CS) pdf divergence measure, and the integrated squared error divergence measure. These will be discussed in more detail in section 4. Since these measures all include quantities which are expressed as integrals over products and squares of densities, we will refer to them as quadratic information measures. Information theoretic learning based on the quadratic information measures, combined with Parzen windowing, has been applied with great success by Principe and co-workers on several supervised and unsupervised learning problems (Lazaro et al., 2005, Erdogmus et al., 2004b,a, Erdogmus and Principe, 2003, Erdogmus et al., 2002, Santamaria et al., 2002, Erdogmus and Principe, 2002b,a, Erdogmus et al., 2002, Principe et al., 2000b).

Information theoretic methods have the advantage over Mercer kernel-based methods that they are easier to interpret. Also, the information theoretic measures can be estimated using Parzen windowing. Parzen windowing is a well established density estimation technique, which has been studied since the 1960's. The strengths and weaknesses of the method are well understood. Moreover, techniques for determining a proper *data driven* size for the Parzen window have been thoroughly studied (Parzen, 1962, Devroye, 1989, Silverman, 1986, Scott, 1992, Wand and Jones, 1995).

In this paper, we will show some equivalences between these two learning schemes, which until now has been treated separately. Specifically, we show that Parzen window-based estimators for the quadratic information measures have a dual interpretation as Mercer kernel-based measures, where they are expressed as functions of *mean values* in the Mercer kernel feature space. The Mercer kernel and the Parzen window are shown to be equivalent. This means that if the Parzen window size can be reliably determined, then the corresponding Mercer kernel size is simultaneously determined by the same procedure.

Furthermore, we develop a classification rule based on the integrated squared error measure, and show that this corresponds to a linear classifier in the feature space. By regarding this classifier as a special case of the support vector machine, we provide an information theoretic interpretation of the SVM optimization criterion.

This paper is organized as follows. In section 2, we review the idea behind Mercer kernel-based learning theory. In section 3, we give a brief review of the SVM. We discuss the Parzen window-based estimators for the quadratic information measures in section 4,

and show the relationship to Mercer kernel feature space quantities. The new information theoretic classification rule is derived in section 5. Thereafter, we analyze the connection between this classifier and the SVM in section 6. We make our concluding remarks in section 7.

2. Mercer Kernel-Based Learning Theory

Mercer kernel-based learning algorithms make use of the following idea: via a nonlinear mapping

$$\begin{aligned} \mathbf{\Phi} : R^d &\rightarrow \mathcal{F} \\ \mathbf{x} &\rightarrow \mathbf{\Phi}(\mathbf{x}) \end{aligned} \tag{1}$$

the data $\mathbf{x}_1, \ldots, \mathbf{x}_N \in R^d$ is mapped into a potentially much higher dimensional feature space $\mathcal{F}$. For a given learning problem one now considers the same learning problem in $\mathcal{F}$ instead of in R^d, working with $\mathbf{\Phi}(\mathbf{x}_1), \ldots, \mathbf{\Phi}(\mathbf{x}_N) \in \mathcal{F}$.

The learning algorithm itself is typically linear in nature, and can be expressed solely in terms of inner-product evaluations. This makes it possible to apply the algorithm in feature space without actually carrying out the data mapping. The key ingredient is a highly effective trick for computing inner products in the feature space using *kernel functions*. One therefore *implicitly* executes the linear algorithm in kernel feature space. This property is advantageous since execution of the learning algorithm in a very high dimensional space is avoided. Because of the non-linear data mapping, the linear operation in kernel feature space corresponds to a non-linear operation in the input space.

Consider a symmetric kernel function $k(\mathbf{x}, \mathbf{y})$. If $k : \mathcal{C} \times \mathcal{C} \rightarrow R$ is a continuous kernel of a positive integral operator in a Hilbert space $L_2(\mathcal{C})$ on a compact set $\mathcal{C} \in R^d$, i.e.

$$\forall \psi \in L_2(\mathcal{C}) : \int_{\mathcal{C}} k(\mathbf{x}, \mathbf{y}) \psi(\mathbf{x}) \psi(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0. \tag{2}$$

Then there exists a space $\mathcal{F}$ and a mapping $\mathbf{\Phi} : R^d \rightarrow \mathcal{F}$, such that by Mercer's theorem (Mercer, 1909)

$$k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{y}) \rangle = \sum_{i=1}^{N_{\mathcal{F}}} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{y}), \tag{3}$$

where $\langle \cdot, \cdot \rangle$ denotes an inner product, the ϕ_i's are the eigenfunctions of the kernel and $N_{\mathcal{F}} \leq \infty$ (Müller et al., 1997, Vapnik, 1995). This operation is known as the "kernel-trick", and it implicitly computes an inner-product in the kernel feature space via $k(\mathbf{x}, \mathbf{y})$.

Indeed, it has been pointed out that the kernel trick can be used to develop non-linear generalizations to any algorithm that can be cast in terms of inner-products (Schölkopf et al., 1998, Schölkopf and Smola, 2002). For example, KPCA, KFDA and kernel K-means (Schölkopf et al., 1998, Girolami, 2002a, Dhillon et al., 2004) are simply extensions of the corresponding linear algorithms by applying the kernel-trick on every inner-product evaluation.

A kernel which satisfies Eq. (2) is known as a Mercer kernel. The most widely used Mercer kernel is the radial-basis-function (RBF)

$$k(\mathbf{x}, \mathbf{y}) = \exp\left\{-\frac{||\mathbf{x}-\mathbf{y}||^2}{2\sigma^2}\right\}, \tag{4}$$

where σ is a scale parameter which controls the width of the RBF. A RBF kernel corresponds to an infinite-dimensional Mercer kernel feature space, since the RBF has an infinite number of eigenfunctions.

3. The Support Vector Machine

The support vector machine is the most prominent Mercer kernel-based learning algorithm. It is a hyperplane classifier which is based on two crucial operations: 1) The kernel-trick, which makes the otherwise linear SVM algorithm non-linear. 2) The maximization of the hyperplane margin, which is a *regularizing* condition on the hyperplane solution. Basically, it limits the admissible separating hyperplanes to the one maximizing the margin. This regularization has a positive effect on the generalization capability of the classifier (Vapnik, 1995).

In the following, we give a brief review of the SVM theory. We formulate the problem directly in the Mercer kernel feature space. This Mercer kernel feature space is induced by some kernel function, which hopefully makes the feature space data linearly separable such that it can be separated by a hyperplane. Whether or not the data in fact is linearly separable, heavily depends on the user choosing a proper kernel.

Let ω_1 and ω_2 denote two data classes. We are given a training set consisting of $\{\mathbf{x}_i\},\ i = 1, \ldots, N_1$, from ω_1, and $\{\mathbf{x}_j\},\ j = 1, \ldots, N_2$, from ω_2. The task is to train a SVM classifier, such that it creates a maximum margin linear classifier in the kernel feature space. After training, the classification rule in feature space is

$$\mathbf{x}_0 \rightarrow \omega_1 : \quad {\mathbf{w}^*}^T \mathbf{\Phi}(\mathbf{x}_0) + b^* \geq 0, \tag{5}$$

otherwise, $\mathbf{x}_0 \rightarrow \omega_2$. Here, $\mathbf{x}_0$ is a new, previously unseen data point. Presumably, it has either been generated by the process generating the ω_1 data, or the process generating the ω_2 data.

Regularizing by maximizing the margin in feature space corresponds to *minimizing the squared norm* of the (canonical) separating hyperplane weight vector, that is $||\mathbf{w}^*||^2$, given the constraints

$$\begin{aligned} {\mathbf{w}^*}^T \mathbf{\Phi}(\mathbf{x}_i) + b^* &\geq +1, \quad \forall \mathbf{x}_i \in \omega_1 \\ {\mathbf{w}^*}^T \mathbf{\Phi}(\mathbf{x}_j) + b^* &\leq -1, \quad \forall \mathbf{x}_j \in \omega_2. \end{aligned} \tag{6}$$

This is a constrained optimization problem, which is dealt with by introducing Lagrange multipliers $\alpha_i \geq 0$, $\alpha_j \geq 0$, corresponding to the two classes, and a primal Lagrangian

$$L_P = \frac{1}{2}||\mathbf{w}^*||^2 - \sum_{i=1}^{N_1} \alpha_i [{\mathbf{w}^*}^T \mathbf{\Phi}(\mathbf{x}_i) + b^* - 1] + \sum_{j=1}^{N_2} \alpha_j [{\mathbf{w}^*}^T \mathbf{\Phi}(\mathbf{x}_j) + b^* + 1]. \tag{7}$$

The Lagrangian L_P has to be minimized with respect to the primal variables $\mathbf{w}^*$ and b^*, and maximized with respect to the *dual* variables $\alpha_i,\ \alpha_j$. Hence a saddle point must be found. At the saddle point, the derivatives of L_P with respect to the primal variables must vanish,

$$\frac{\partial}{\partial b^*}L_P = 0, \quad \frac{\partial}{\partial \mathbf{w}^*}L_P = 0, \tag{8}$$

which leads to

$$\sum_{i=1}^{N_1}\alpha_i = \sum_{j=1}^{N_2}\alpha_j = \Omega, \tag{9}$$

and

$$\mathbf{w}^* = \mathbf{m}_1^* - \mathbf{m}_2^*, \tag{10}$$

where

$$\mathbf{m}_1^* = \sum_{i=1}^{N_1}\alpha_i\mathbf{\Phi}(\mathbf{x}_i), \quad \mathbf{m}_2^* = \sum_{j=1}^{N_2}\alpha_j\mathbf{\Phi}(\mathbf{x}_j). \tag{11}$$

By substituting these constraints into Eq. (7), the dual Lagrangian

$$L_D = 2\Omega - \frac{1}{2}\left\{\sum_{i,i'=1}^{N_1,N_1}\alpha_i\alpha_{i'}k(\mathbf{x}_i,\mathbf{x}_{i'}) - 2\sum_{i,j=1}^{N_1,N_2}\alpha_i\alpha_j k(\mathbf{x}_i,\mathbf{x}_j) + \sum_{j,j'=1}^{N_2,N_2}\alpha_j\alpha_{j'}k(\mathbf{x}_j,\mathbf{x}_{j'})\right\}, \tag{12}$$

is obtained, where $k(\cdot,\cdot)$ denotes an inner product between any two training data points in the Mercer kernel feature space. L_D must be maximized with respect to the Lagrange multipliers. It can be seen that the solution vector $\mathbf{w}^*$ has an expansion in terms of the training patterns weighted by the Lagrange multipliers. The Karush-Kuhn-Tucker (KKT) conditions

$$\begin{aligned}\alpha_i[{\mathbf{w}^*}^T\mathbf{\Phi}(\mathbf{x}_i) + b^* - 1] &= 0, \quad \forall i = 1,\ldots,N_1,\\ \alpha_j[{\mathbf{w}^*}^T\mathbf{\Phi}(\mathbf{x}_j) + b^* + 1] &= 0, \quad \forall j = 1,\ldots,N_2,\end{aligned} \tag{13}$$

specify the *non-zero* Lagrange multipliers to be those training patterns which are situated on the margin in feature space. Hence, $\mathbf{w}^*$ is a weighted combination of the patterns on the margin.

Let us determine the expression for b^* in the SVM theory. For those b^* corresponding to support vectors belonging to ω_1, we have $b_1^* = 1 - {\mathbf{w}^*}^T\mathbf{\Phi}(\mathbf{x}_i)$, where $\mathbf{\Phi}(\mathbf{x}_i)$ is a support vector. By adding all b_1^* values corresponding to ω_1, we have (remember that only those α_i's corresponding to support vectors deviate from zero)

$$\begin{aligned}\sum_{i=1}^{N_1}\alpha_i b_1^* &= \sum_{i=1}^{N_1}\alpha_i - {\mathbf{w}^*}^T\sum_{i=1}^{N_1}\alpha_i\mathbf{\Phi}(\mathbf{x}_i)\\ \Omega b_1^* &= \Omega - {\mathbf{w}^*}^T\mathbf{m}_1^*\\ b_1^* &= 1 - \frac{1}{\Omega}||\mathbf{m}_1^*||^2 + \frac{1}{\Omega}{\mathbf{m}_1^*}^T\mathbf{m}_2^*.\end{aligned} \tag{14}$$

Similarly, for those b^* corresponding to support vectors belonging to ω_2, we have $b_2^* = -1 - \mathbf{w}^{*^T}\mathbf{\Phi}(\mathbf{x}_j)$. Again, by adding all b_2^* corresponding to ω_2, we obtain

$$\begin{aligned}\sum_{j=1}^{N_2}\alpha_j b_2^* &= -\sum_{j=1}^{N_2}\alpha_j - \mathbf{w}^{*^T}\sum_{j=1}^{N_2}\alpha_j\mathbf{\Phi}(\mathbf{x}_j)\\ \Omega b_2^* &= -\Omega - \mathbf{w}^{*^T}\mathbf{m}_2^*\\ b_2^* &= -1 - \frac{1}{\Omega}\mathbf{m}_1^{*^T}\mathbf{m}_2^* + \frac{1}{\Omega}||\mathbf{m}_2^*||^2. \end{aligned} \tag{15}$$

Since $b_1^* = b_2^*$, we have $b^* = \frac{1}{2}[b_1^* + b_2^*]$, such that

$$b^* = \frac{1}{2\Omega}[||\mathbf{m}_2^*||^2 - ||\mathbf{m}_1^*||^2]. \tag{16}$$

4. Quadratic Information Measures and Parzen Windowing

In this section, we will review the quadratic information measures, and show how they may be estimated non-parametrically using the Parzen window technique for density estimation. For details on how these cost functions may be used in adaptive systems training, we refer to (Principe et al., 2000a,b). We will also show how each of these measures can be expressed in terms of *mean values* in a Mercer kernel feature space.

4.1 Parzen Window Density Estimator

Parzen windowing is a well-known kernel-based density estimation method (Devroye and Lugosi, 2001, Parzen, 1962). Given a set of iid samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ drawn from the true density $f(\mathbf{x})$, the Parzen window estimator for this distribution is defined as

$$\hat{f}(\mathbf{x}) = \frac{1}{N}\sum_{t=1}^{N} W_{\sigma^2}(\mathbf{x}, \mathbf{x}_t). \tag{17}$$

Here, W_{σ^2} is the Parzen window, or kernel, and σ^2 controls the width of the kernel. The Parzen window must integrate to one, and is typically chosen to be a pdf itself, such as the Gaussian kernel. Hence,

$$W_{\sigma^2}(\mathbf{x}, \mathbf{x}_t) = \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}}\exp\left\{-\frac{||\mathbf{x} - \mathbf{x}_t||^2}{2\sigma^2}\right\}.$$

We will use the Gaussian kernel in the derivations that follows, but show in the Appendix that other choices may be used. Note also that the *width* of the Parzen window affects the density estimate much more than the actual form of the window function (Scott, 1992, Wand and Jones, 1995).

It is easily shown that Eq. (17) is an asymptotically unbiased and consistent estimator, provided σ decays to zero at a certain rate as N tends to infinity (Parzen, 1962). In the finite sample case, the window width is usually chosen such that it minimizes the mean integrated squared error (MISE) between $\hat{f}(\mathbf{x})$ and the target density $f(\mathbf{x})$. It is easily

shown that the MISE consists of a bias part and a variance part. Unfortunately, the bias part is minimized by minimizing the window width, while the variance is minimized by maximizing the window width. This is the inherent bias-variance trade-off associated with the Parzen window technique.

Finding a window width, or kernel size, which provides a good bias-variance trade-off has been thoroughly studied in the statistics literature (Silverman, 1986, Scott, 1992, Wand and Jones, 1995). Especially for data sets of low to moderate dimensionality, many reliable methods exist, such as for example least-squares cross-validation (Wand and Jones, 1995). Another straight-forward and popular approach is to find the kernel size which minimizes the asymptotic MISE (AMISE). By assuming that the underlying density is Gaussian, this kernel size is given by

$$\sigma_{\text{AMISE}} = \sigma_X \left[\frac{4}{(2d+1)N} \right]^{\frac{1}{d+4}}, \tag{18}$$

where $\sigma_X^2 = d^{-1} \sum_i \mathbf{\Sigma}_{X_{ii}}$, and $\mathbf{\Sigma}_{X_{ii}}$ are the diagonal elements of the sample covariance matrix (Silverman, 1986). The main appeal of this approach is that is is very easy to use. The obvious drawback is that it assumes that the underlying density is unimodal and Gaussian. Many other methods exist, each having specific properties.

For high-dimensional data sets, the Parzen window technique for density estimation is known to have severe limitations. The reason is that the usual bias-variance trade-off cannot be accomplished very well in higher dimensions without very large samples (Scott, 1992, Silverman, 1986). This is known as the "curse-of-dimensionality".

Note however that this limitation may not apply directly when the Parzen window technique is used in clustering or classification, as discussed by Friedman (1997). He showed that in those applications, low variance is much more important than low bias, hence favoring a large kernel size.

4.2 Renyi Quadratic Entropy

The Renyi quadratic entropy associated with the pdf $f(\mathbf{x})$ is given by (Renyi, 1976b,a)

$$H_{R_2}(f) = -\log \int f^2(\mathbf{x}) d\mathbf{x}. \tag{19}$$

We have available a sample from $f(\mathbf{x})$, namely $\{\mathbf{x}_t\},\ t = 1, \ldots, N$. Based on the sample, we estimate $f(\mathbf{x})$ by $\hat{f}(\mathbf{x})$, the Parzen window estimator. We obtain an estimate for the Renyi entropy using the *plug-in* a density estimator principle, by replacing $f(\mathbf{x})$ by $\hat{f}(\mathbf{x})$. However, since the logarithm is a monotonic function, we will focus on the quantity $V(f) = \int \hat{f}^2(\mathbf{x}) d\mathbf{x}$, thus given by [1]

$$\begin{aligned} V(f) &= \int \frac{1}{N} \sum_{t=1}^{N} W_{\sigma^2}(\mathbf{x}, \mathbf{x}_t) \frac{1}{N} \sum_{t'=1}^{N} W_{\sigma^2}(\mathbf{x}, \mathbf{x}_{t'}) d\mathbf{x} \\ &= \frac{1}{N^2} \sum_{t,t'=1}^{N,N} \int W_{\sigma^2}(\mathbf{x}, \mathbf{x}_t) W_{\sigma^2}(\mathbf{x}, \mathbf{x}_{t'}) d\mathbf{x}. \end{aligned} \tag{20}$$

1. $\sum_{t,t'=1}^{N,N}$ equals the double summation $\sum_{t=1}^{N} \sum_{t'=1}^{N}$.

Now a property of Gaussian functions is employed. By the convolution theorem for Gaussians, we have

$$\int W_{\sigma^2}(\mathbf{x}, \mathbf{x}_t) W_{\sigma^2}(\mathbf{x}, \mathbf{x}_{t'}) d\mathbf{x} = W_{2\sigma^2}(\mathbf{x}_t, \mathbf{x}_{t'}), \tag{21}$$

that is, the convolution of two Gaussians is a new Gaussian function having twice the (co)variance. Thus, we have

$$V(f) = \frac{1}{N^2} \sum_{t,t'=1}^{N,N} W_{2\sigma^2}(\mathbf{x}_t, \mathbf{x}_{t'}). \tag{22}$$

It can be seen that this estimation procedure involves no approximations, besides the pdf estimator itself. Eq. (22) was named the *information potential* (Principe et al., 2000a), because of an analogy to a potential energy field.

The key point in the following discussion is to note is that $W_{2\sigma^2}(\mathbf{x}_t, \mathbf{x}_{t'})$, for any $\mathbf{x}_t,\ \mathbf{x}_{t'}$, is a Gaussian kernel function, and hence it is also a *kernel function that satisfies Mercer's theorem*. Thus

$$W_{2\sigma^2}(\mathbf{x}_t, \mathbf{x}_{t'}) = k(\mathbf{x}_t, \mathbf{x}_{t'}) = \langle \mathbf{\Phi}(\mathbf{x}_t), \mathbf{\Phi}(\mathbf{x}_{t'}) \rangle . \tag{23}$$

Hence, the Parzen window-based estimator for the information potential can be expressed in terms of inner products in a Mercer kernel space. In the following we make this connection explicit. We rewrite Eq. (20) as follows

$$\begin{aligned} V(f) &= \frac{1}{N^2} \sum_{t,t'=1}^{N,N} \langle \mathbf{\Phi}(\mathbf{x}_t), \mathbf{\Phi}(\mathbf{x}_{t'}) \rangle \\ &= \left\langle \frac{1}{N} \sum_{t=1}^{N} \mathbf{\Phi}(\mathbf{x}_t), \frac{1}{N} \sum_{t'=1}^{N} \mathbf{\Phi}(\mathbf{x}_{t'}) \right\rangle \\ &= \mathbf{m}^T \mathbf{m} \\ &= ||\mathbf{m}||^2, \end{aligned} \tag{24}$$

where $\mathbf{m}$ is the mean vector of the $\mathbf{\Phi}$-transformed data

$$\mathbf{m} = \frac{1}{N} \sum_{t=1}^{N} \mathbf{\Phi}(\mathbf{x}_t). \tag{25}$$

That is, it turns out that the information potential may be expressed as the squared norm of the *mean vector* of the data in a Mercer kernel feature space. This connection was previously pointed out by Girolami (2002b) in a study relating orthogonal series density estimation to kernel principal component analysis.

4.3 Integrated Squared Error as a PDF Divergence

In order to measure the "distance" or divergence between two probability densities, $p(\mathbf{x})$ and $q(\mathbf{x})$, Principe et al. (2000a) proposed an integrated squared error (ISE) criterion. The

$ISE(p,q)$ is given by (Principe et al., 2000a)

$$\begin{aligned} ISE(p,q) &= \int [p(\mathbf{x}) - q(\mathbf{x})]^2 \, d\mathbf{x} \\ &= \int p^2(\mathbf{x}) d\mathbf{x} - 2\int p(\mathbf{x})q(\mathbf{x}) d\mathbf{x} + \int q^2(\mathbf{x}) d\mathbf{x}. \end{aligned} \tag{26}$$

It can be seen that the integrated squared error criterion is always non-negative, it is symmetric and it vanishes if and only if the two pdfs are identical.

In this case, we have available a sample from $p(\mathbf{x})$, namely $\{\mathbf{x}_i\},\ i = 1, \ldots, N_1$, and a corresponding sample from $q(\mathbf{x})$, that is, $\{\mathbf{x}_j\},\ j = 1, \ldots, N_2$[2]. We estimate the two pdfs by the Parzen window method

$$\hat{p}(\mathbf{x}) = \frac{1}{N_1}\sum_{i=1}^{N_1} W_{\sigma^2}(\mathbf{x}, \mathbf{x}_i), \quad \hat{q}(\mathbf{x}) = \frac{1}{N_2}\sum_{j=1}^{N_2} W_{\sigma^2}(\mathbf{x}, \mathbf{x}_j). \tag{27}$$

These estimators are now used to estimate the ISE divergence. Note that we have for simplicity assumed that the same kernel size σ is appropriate for both estimators. This may not be the case in practice. The latter situation may easily be incorporated in the subsequent analysis. Now, performing a similar calculation as above, the ISE can be estimated non-parametrically as follows

$$\widehat{ISE}(p,q) = \frac{1}{N_1^2}\sum_{i,i'=1}^{N_1,N_1} W_{2\sigma^2}(\mathbf{x}_i, \mathbf{x}_{i'}) - \frac{2}{N_1 N_2}\sum_{i,j=1}^{N_1,N_2} W_{2\sigma^2}(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{N_2^2}\sum_{j,j'=1}^{N_2,N_2} W_{2\sigma^2}(\mathbf{x}_j, \mathbf{x}_{j'}).$$

In analogy to Eq. (22), the $\widehat{ISE}$ may also be expressed in terms of mean vectors in the Mercer kernel feature space. When we perform a similar calculation, we obtain

$$\begin{aligned} \widehat{ISE}(p,q) &= ||\mathbf{m}_1||^2 - 2\mathbf{m}_1^T\mathbf{m}_2 + ||\mathbf{m}_2||^2 \\ &= ||\mathbf{m}_1 - \mathbf{m}_2||^2, \end{aligned} \tag{28}$$

where $\mathbf{m}_1$ is the kernel feature space mean vector of the data points drawn from $p(\mathbf{x})$, and $\mathbf{m}_2$ is the kernel feature space mean vector of the data points drawn from $q(\mathbf{x})$. That is

$$\mathbf{m}_1 = \frac{1}{N_1}\sum_{i=1}^{N_1} \mathbf{\Phi}(\mathbf{x}_i) \quad \mathbf{m}_1 = \frac{1}{N_2}\sum_{j=1}^{N_2} \mathbf{\Phi}(\mathbf{x}_j). \tag{29}$$

Hence, the ISE divergence measure can also be seen to have a geometric interpretation in the kernel feature space. It measures the square of the norm of the difference vector between the two means $\mathbf{m}_1$ and $\mathbf{m}_2$.

2. In the following, the index i always points to data drawn from $p(\mathbf{x})$, j always points to data drawn from $q(\mathbf{x})$, and t always points to data drawn from $f(\mathbf{x})$.

4.4 Cauchy-Schwarz PDF Divergence

Based on the Cauchy-Schwarz inequality, Principe et al. (2000a) also proposed the following divergence measure between the pdfs $p(\mathbf{x})$ and $q(\mathbf{x})$

$$D_{CS}(p,q) = -\log \frac{\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}}{\sqrt{\int p^2(\mathbf{x})d\mathbf{x} \int q^2(\mathbf{x})d\mathbf{x}}}, \tag{30}$$

which we refer to as the Cauchy-Schwarz pdf divergence. It is also always non-negative, it is symmetric and it vanishes if and only if the two densities are equal.

Since the logarithm is monotonic, we will focus on the quantity in the argument of the $\log$ in Eq. (30). The Parzen window-based estimator for this quantity was named the *information cut* (IC) in (Jenssen et al., 2003), because it was shown to be closely related to the graph theoretic *cut*. By a similar calculation as above, the IC can be expressed as

$$IC(p,q) = \frac{\frac{1}{N_1 N_2}\sum_{i,j=1}^{N_1,N_2} W_{2\sigma^2}(\mathbf{x}_i,\mathbf{x}_j)}{\sqrt{\frac{1}{N_1^2}\sum_{i,i'=1}^{N_1,N_1} W_{2\sigma^2}(\mathbf{x}_i,\mathbf{x}_{i'})\ \frac{1}{N_2^2}\sum_{j,j'=1}^{N_2,N_2} W_{2\sigma^2}(\mathbf{x}_j,\mathbf{x}_{j'})}}. \tag{31}$$

Also, the information cut may be expressed in terms of mean vectors in the Mercer kernel feature space, as

$$IC = \frac{\mathbf{m}_1^T \mathbf{m}_2}{||\mathbf{m}_1||||\mathbf{m}_2||} = \cos\angle(\mathbf{m}_1,\mathbf{m}_2), \tag{32}$$

Hence, it turns out that the information cut has a dual interpretation as a measure of the cosine of the angle between cluster mean vectors in the Mercer kernel feature space.

5. ISE-Based Classification

In this section we will propose a new classification rule based on the ISE, which we will analyze theoretically both in the input space and in the Mercer kernel space. An interesting property of this new classifier is that is contains the Bayes classifier as a special case.

We have available the training data points $\{\mathbf{x}_i\},\ i = 1,\ldots,N_1$, drawn from $p(\mathbf{x})$, and a corresponding sample from $q(\mathbf{x})$, that is, $\{\mathbf{x}_j\},\ j = 1,\ldots,N_2$. Based on this training data set we wish to construct a classifier, which assigns a test data point $\mathbf{x}_0$ to one of the classes ω_1 or ω_2. Now, we define

$$\hat{p}'(\mathbf{x}) = \frac{1}{N_1+1}\sum_{i=0}^{N_1} W_{\sigma^2}(\mathbf{x},\mathbf{x}_i), \quad \hat{q}'(\mathbf{x}) = \frac{1}{N_2+1}\sum_{j=0}^{N_2} W_{\sigma^2}(\mathbf{x},\mathbf{x}_j). \tag{33}$$

Hence, $\hat{p}'(\mathbf{x})$ is the Parzen estimator for $p(\mathbf{x})$, assuming $\mathbf{x}_0$ is included in the ω_1 data set. Likewise, $\hat{q}'(\mathbf{x})$ is the Parzen estimator for $q(\mathbf{x})$, assuming $\mathbf{x}_0$ is included in the ω_2 data set.

The proposed ISE-based strategy is to classify $\mathbf{x}_0$ according to the following rule:

$$\mathbf{x}_0 \rightarrow \omega_1 : \int [\hat{p}'(\mathbf{x}) - \hat{q}(\mathbf{x})]^2 d\mathbf{x} \geq \int [\hat{p}(\mathbf{x}) - \hat{q}'(\mathbf{x})]^2 d\mathbf{x}, \tag{34}$$

otherwise, assign $\mathbf{x}_0$ to ω_2. In words; assign $\mathbf{x}_0$ to the class which, when having $\mathbf{x}_0$ appended to it, makes the estimated divergence between the classes the greatest.

We will now analyze this simple classification rule in terms of the Mercer kernel feature space. Let $\mathbf{m}_i', i = 1, 2$ be the Mercer kernel feature space mean vector of class ω_i, assuming $\mathbf{\Phi}(\mathbf{x}_0)$ is assigned to that class. It is easily shown that

$$\begin{aligned} \mathbf{m}_1' &= \frac{N_1}{N_1+1}\mathbf{m}_1 + \frac{1}{N_1+1}\mathbf{\Phi}(\mathbf{x}_0) \\ \mathbf{m}_2' &= \frac{N_2}{N_2+1}\mathbf{m}_2 + \frac{1}{N_2+1}\mathbf{\Phi}(\mathbf{x}_0). \end{aligned} \tag{35}$$

In the kernel feature space, the equivalent classification rule of Eq. (34) may be expressed as

$$\mathbf{x}_0 \rightarrow \omega_1 : \quad ||\mathbf{m}_1' - \mathbf{m}_2||^2 \geq ||\mathbf{m}_1 - \mathbf{m}_2'||^2. \tag{36}$$

In what follows, we look at a special case. *Assume that $P(\omega_1) = P(\omega_2)$, that is the prior probabilities for the classes are equal. Let $P(\omega_1) = \frac{N_1}{N}$ and $P(\omega_2) = \frac{N_2}{N}$, which means that we assume that $N_1 = N_2$.* In that case, we have

$$\begin{aligned} \mathbf{m}_1' &= \kappa_1\mathbf{m}_1 + \kappa_2\mathbf{\Phi}(\mathbf{x}_0) \\ \mathbf{m}_2' &= \kappa_1\mathbf{m}_2 + \kappa_2\mathbf{\Phi}(\mathbf{x}_0), \end{aligned} \tag{37}$$

where $\kappa_1 = \frac{N_1}{N_1+1} = \frac{N_2}{N_2+1}$, and $\kappa_2 = \frac{1}{N_1+1} = \frac{1}{N_2+1}$.

For ease of notation, let $\mathbf{\Phi}(\mathbf{x}_0) = \mathbf{y}$. The left-hand side of Eq.(36), becomes

$$\begin{aligned} ||\mathbf{m}_1' - \mathbf{m}_2||^2 &= \mathbf{m}_1'^T\mathbf{m}_1' - 2\mathbf{m}_1'^T\mathbf{m}_2 + \mathbf{m}_2^T\mathbf{m}_2 \\ &= \kappa_1^2||\mathbf{m}_1||^2 + 2\kappa_1\kappa_2\mathbf{m}_1^T\mathbf{y} + \kappa_2^2||\mathbf{y}||^2 - 2\kappa_1\mathbf{m}_1^T\mathbf{m}_2 - 2\kappa_2\mathbf{m}_2^T\mathbf{y} + ||\mathbf{m}_2||^2. \end{aligned}$$

Similarly, the right-hand side of Eq.(36) becomes

$$\begin{aligned} ||\mathbf{m}_1 - \mathbf{m}_2'||^2 &= \mathbf{m}_1^T\mathbf{m}_1 - 2\mathbf{m}_1^T\mathbf{m}_2' + \mathbf{m}_2'^T\mathbf{m}_2 \\ &= ||\mathbf{m}_1||^2 - 2\kappa_1\mathbf{m}_2^T\mathbf{m}_1 - 2\kappa_2\mathbf{m}_1^T\mathbf{y} + \kappa_1^2||\mathbf{m}_2||^2 + 2\kappa_1\kappa_2\mathbf{m}_2^T\mathbf{y} + \kappa_2^2||\mathbf{y}||^2. \end{aligned}$$

Using these results, the classification rule becomes

$$\begin{aligned} \mathbf{x}_0 \rightarrow \omega_1 \quad : \quad & ||\mathbf{m}_1' - \mathbf{m}_2||^2 \geq ||\mathbf{m}_1 - \mathbf{m}_2'||^2 \\ \Leftrightarrow \quad & \mathbf{m}_1^T\mathbf{y} - \mathbf{m}_2^T\mathbf{y} - \frac{\kappa_1^2 - 1}{2\kappa_2[\kappa_1+1]}\left[||\mathbf{m}_2||^2 - |\mathbf{m}_1||^2\right] \geq 0 \\ \Leftrightarrow \quad & \mathbf{m}_1^T\mathbf{y} - \mathbf{m}_2^T\mathbf{y} + b \geq 0. \end{aligned} \tag{38}$$

where $b = \frac{1}{2}\left[||\mathbf{m}_2||^2 - ||\mathbf{m}_1||^2\right]$, and the constant $\frac{\kappa_1^2-1}{\kappa_2[\kappa_1+1]} = -1$.

In fact, the above classification rule has a simple geometrical interpretation. *The point $\mathbf{y}$ is assigned to the class whose mean it is closest, and the class boundary in kernel feature space is a hyperplane given by a vector $\mathbf{w}$.* Let $\mathbf{w} = \mathbf{m}_1 - \mathbf{m}_2$, and let the midpoint between $\mathbf{m}_1$ and $\mathbf{m}_2$ be given by $\mathbf{v} = \frac{1}{2}(\mathbf{m}_1 + \mathbf{m}_2)$. Now the class of $\mathbf{y}$ is determined by examining whether

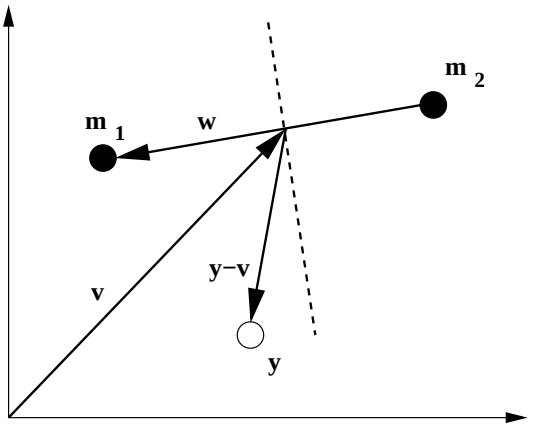


Figure 2: ISE-based geometric classification rule: Assign the point $\mathbf{y}$ to the class whose mean it is closest to. This can be done by looking at the inner-product between $(\mathbf{y}-\mathbf{v})$ and $\mathbf{w}$. It changes sign as the enclosed angle passes through $\frac{\pi}{2}$. The corresponding decision boundary is given by a *hyperplane* orthogonal to $\mathbf{w}$ (dashed line).

the vector $(\mathbf{y}-\mathbf{v})$ encloses an angle smaller than $\frac{\pi}{2}$ with the vector $\mathbf{w}$ or not. If it does, $\mathbf{y}$ is closest to $\mathbf{m}_1$, and $\mathbf{y}$ is assigned to ω_1. Hence,

$$\begin{aligned} \mathbf{x}_0 \rightarrow \omega_1 \quad &: \quad \mathbf{w}^T(\mathbf{y}-\mathbf{v}) \geq 0 \\ &\Leftrightarrow \quad \mathbf{w}^T\mathbf{y} + b \geq 0 \\ &\Leftrightarrow \quad \mathbf{m}_1^T\mathbf{y} - \mathbf{m}_2^T\mathbf{y} + b \geq 0, \end{aligned} \tag{39}$$

Figure 2 geometrically illustrates this simple classification rule, which we have derived using the ISE criterion as a starting point.

As explained above, in the Mercer kernel space, the value of the inner-product between the class mean values and the new data point determines which class it is assigned to. The threshold value, b, depends on the squared Euclidean norms of the mean values, which according to Eq. (24) are equivalent to the class information potentials, and hence the class entropies.

We now complete the circle, and analyze the Mercer kernel feature space classification rule in terms of Parzen estimators in the input space. Note that

$$\mathbf{m}_1^T\mathbf{y} = \mathbf{m}_1^T\mathbf{\Phi}(\mathbf{x}_0) = \frac{1}{N_1}\sum_{i=1}^{N_1}\mathbf{\Phi}^T(\mathbf{x}_i)\mathbf{\Phi}(\mathbf{x}_0) = \frac{1}{N_1}\sum_{i=1}^{N_1}W_{\sigma^2}(\mathbf{x}_0,\mathbf{x}_i) = \hat{p}(\mathbf{x}_0). \tag{40}$$

Likewise

$$\mathbf{m}_2^T\mathbf{y} = \mathbf{m}_2^T\mathbf{\Phi}(\mathbf{x}_0) = \frac{1}{N_2}\sum_{j=1}^{N_2}\mathbf{\Phi}^T(\mathbf{x}_j)\mathbf{\Phi}(\mathbf{x}_0) = \frac{1}{N_2}\sum_{j=1}^{N_2}W_{\sigma^2}(\mathbf{x}_0,\mathbf{x}_j) = \hat{q}(\mathbf{x}_0). \tag{41}$$

The classification rule hence becomes

$$\mathbf{x}_0 \rightarrow \omega_1 : \quad \hat{p}(\mathbf{x}_0) - \hat{q}(\mathbf{x}_0) + b \geq 0. \tag{42}$$

We remark that this new classification rule depends both on the estimated densities at $\mathbf{x}_0$, and on the information potentials of the two classes. We have already shown that these information potentials are equivalent to Renyi's quadratic entropies for the classes.

In the case that the classes have the same value for the information potential (entropy), which means that the kernel feature space mean values have equal length from the origin, we have $b = 0$, and the current classification rule reduces to the well-known Bayes classification rule (for equal priors), where the class probability densities are estimated using Parzen windowing.

The same direct connection can not be obtained based on the Cauchy-Schwarz divergence.

6. ISE-Based Classification and the SVM

In the previous section, we derived a new type of information theoretic classification rule, which turned out to have a dual interpretation as a *hyperplane classifier* in a Mercer kernel feature space. This classifier is entirely determined by the mean vectors $\mathbf{m}_1$ and $\mathbf{m}_2$ of the training data, since both $\mathbf{w}$ and b are determined by these vectors. For the classifier to perform well on test data, we are totally dependent on these mean vectors to truly represent the structure of the data. For example, the presence of outliers in the training set may affect the computation of $\mathbf{w}$ and b in such a way that the performance of the classifier is degraded. This may be remedied by allowing the contribution of each training data point to the mean vectors to be weighted differently. Let us therefore introduce the weighting components $\alpha_i \geq 0$ associated with ω_1, and $\alpha_j \geq 0$ associated with ω_2. The *weighted mean vectors* then become

$$\mathbf{m}_1 = \frac{1}{\Omega_1} \sum_{i=1}^{N_1} \alpha_i \mathbf{\Phi}(\mathbf{x}_i), \quad \mathbf{m}_2 = \frac{1}{\Omega_2} \sum_{j=1}^{N_2} \alpha_j \mathbf{\Phi}(\mathbf{x}_j). \tag{43}$$

By introducing such weighted mean vectors, we also need to introduce some criterion to determine proper weights. Such a criterion should be optimal with respect to classifier performance. The performance of a classifier is measured by its success rate on test data. Hence, the classifier should generalize well. In statistical learning theory, it has been shown that minimization of the squared norm of the hyperplane weight vector, while satisfying the classification constraints on the training data, improves generalization performance.

Based on the arguments above, we may relate the vector $\mathbf{w} = \mathbf{m}_1 - \mathbf{m}_2$ to the SVM weight vector $\mathbf{w}^* = \mathbf{m}_1^* - \mathbf{m}_2^*$. Recall that the SVM is exactly based on regularization by minimization of $||\mathbf{w}^*||^2$. The minimization is accompanied by the classification constraints, which ensures that the training data is classified correctly. These constraints say

$$\begin{aligned} {\mathbf{w}^*}^T \mathbf{\Phi}(\mathbf{x}_i) + b^* &\geq +1, \quad \forall \mathbf{x}_i \in \omega_1 \\ {\mathbf{w}^*}^T \mathbf{\Phi}(\mathbf{x}_j) + b^* &\leq -1, \quad \forall \mathbf{x}_j \in \omega_2. \end{aligned} \tag{44}$$

In fact, if Ω_1 and Ω_2 were equal, then $\mathbf{w}$ and $\mathbf{w}^*$ would only differ by a constant.

Let us take a closer look at the information potentials associated with the weighted mean vectors. We have

$$||\mathbf{m}_1||^2 = \frac{1}{\Omega_1^2} \sum_{i,i'=1}^{N_1,N_1} \alpha_i \alpha_{i'} k(\mathbf{x}_i, \mathbf{x}_{i'}) = \int \hat{p}^2(\mathbf{x}) d\mathbf{x}. \tag{45}$$

Thus, the weighted mean vector $\mathbf{m}_1$ is associated with

$$\hat{p}(\mathbf{x}) = \frac{1}{\Omega_1} \sum_{i=1}^{N_1} \alpha_i W_{\sigma^2}(\mathbf{x}, \mathbf{x}_i), \tag{46}$$

a weighted Parzen window estimator in the input space. We likewise have

$$\hat{q}(\mathbf{x}) = \frac{1}{\Omega_2} \sum_{j=1}^{N_2} \alpha_j W_{\sigma^2}(\mathbf{x}, \mathbf{x}_j). \tag{47}$$

The kernels which constitute these Parzen window estimators are no longer equally important. Recall that the ISE classification rule based on the density estimators is

$$\mathbf{x}_0 \rightarrow \omega_1 \quad : \qquad \hat{p}(\mathbf{x}_0) - \hat{q}(\mathbf{x}_0) + b \geq 0, \tag{48}$$

with $b = \frac{1}{2}\left[||\mathbf{m}_2||^2 - ||\mathbf{m}_1||^2\right]$. In order to derive this classification rule using the traditional Parzen window estimators, we assumed that $N_1 = N_2$. Using the weighted Parzen window estimators instead, it is easily found that the corresponding assumption becomes $\Omega_1 = \Omega_2 = \Omega$ (see Eq. (37) and the related dicussion in section 5). Therefore,

$$\mathbf{m}_1 = \frac{1}{\Omega}\mathbf{m}_1^*, \quad \mathbf{m}_2 = \frac{1}{\Omega}\mathbf{m}_2^*, \tag{49}$$

and consequently

$$\mathbf{w} = \frac{1}{\Omega}\mathbf{w}^*. \tag{50}$$

Now, using the weighted Parzen window estimators we may express the SVM optimization problem in an *information theoretic framework* as follows

$$\min_{\alpha_i,\, \alpha_j} ||\mathbf{w}^*||^2 = \min_{\alpha_i,\, \alpha_j} \Omega^2 ||\mathbf{w}||^2 = \min_{\alpha_i,\, \alpha_j} \Omega^2 ||\mathbf{m}_1 - \mathbf{m}_2||^2. \tag{51}$$

Since $||\mathbf{m}_1 - \mathbf{m}_2||^2$ is the Mercer kernel feature space equivalent to the ISE pdf divergence, we have

$$\min_{\alpha_i,\, \alpha_j} \Omega^2 ||\mathbf{m}_1 - \mathbf{m}_2||^2 = \min_{\alpha_i,\, \alpha_j} \Omega^2 \int [\hat{p}(\mathbf{x}) - \hat{q}(\mathbf{x})]^2 d\mathbf{x}. \tag{52}$$

The optimization is subject to classification constraints, expressed as

$$\begin{aligned}
&1) \qquad {\mathbf{w}^*}^T \mathbf{\Phi}(\mathbf{x}_i) + b^* \geq 1 \\
&\Leftrightarrow \qquad \Omega \mathbf{w}^T \mathbf{\Phi}(\mathbf{x}_i) + \Omega b \geq 1 \\
&\Leftrightarrow \qquad \mathbf{w}^T \mathbf{\Phi}(\mathbf{x}_i) + b \geq \frac{1}{\Omega} \\
&\Leftrightarrow \qquad \hat{p}(\mathbf{x}_i) - \hat{q}(\mathbf{x}_i) + b \geq \frac{1}{\Omega},
\end{aligned} \tag{53}$$

for $i = 1, \ldots, N_1$.

$$\begin{aligned} & 2) \quad && \mathbf{w}^{*T}\mathbf{\Phi}(\mathbf{x}_j) + b^* \leq -1 \\ \Leftrightarrow \quad & && \Omega\mathbf{w}^T\mathbf{\Phi}(\mathbf{x}_j) + \Omega b \leq -1 \\ \Leftrightarrow \quad & && \mathbf{w}^T\mathbf{\Phi}(\mathbf{x}_j) + b \leq -\frac{1}{\Omega} \\ \Leftrightarrow \quad & && \hat{p}(\mathbf{x}_j) - \hat{q}(\mathbf{x}_j) + b \leq -\frac{1}{\Omega}, \end{aligned} \tag{54}$$

for $j = 1, \ldots, N_2$.

Likewise, the SVM classification rule, using the weighted Parzen window estimators, becomes

$$\begin{aligned} \mathbf{x}_0 \rightarrow \omega_1 \quad & : && \mathbf{w}^{*T}\mathbf{\Phi}(\mathbf{x}_0) + b^* \geq 0 \\ & \Leftrightarrow && \Omega\mathbf{w}^T\mathbf{\Phi}(\mathbf{x}_0) + \Omega b \geq 0 \\ & \Leftrightarrow && \mathbf{w}^T\mathbf{\Phi}(\mathbf{x}_0) + b \geq 0 \\ & \Leftrightarrow && \hat{p}(\mathbf{x}_0) - \hat{q}(\mathbf{x}_0) + b \geq 0. \end{aligned} \tag{55}$$

The weighted Parzen window estimators $\hat{p}(\mathbf{x})$ and $\hat{q}(\mathbf{x})$, as defined above, are *bona fide* density estimators. That is, they are always non-negative and integrate to one. However, since the weights are determined by minimizing the ISE pdf divergence, which puts emphasis on the points close to the class boundary trying to maximize the overlap between the class pdfs, we do not regard them as proper estimators for the pdfs that generated the data. From SVM theory, we know that in the Mercer kernel feature space, the only non-zero weighting components are those which correspond to data patterns on the margin. In the input space, it seems that the corresponding non-zero weighting components will be associated with data patterns near the class boundary. We therefore interpret the minimization of the ISE pdf divergence as a *sparseness criterion*, which tunes the classifier to those patterns which are near the boundary. The other data patterns should be much easier to classify correctly, and are not given any weight in the design of the classifier. The performance of the classifier is secured by the classification constraints. Note that weighted Parzen window estimators have been previously proposed for improved Parzen window-based Bayes classification (Babich and Camps, 1996, Marzio and Taylor, 2004).

In summary, we have found that one may view the SVM theory in feature space in terms of weighted Parzen density estimation in the input space, where regularization is obtained by minimizing the integrated squared error criterion. Hence, in an information theoretic framework, the support vector machine is formulated by introducing the weights $\alpha_i \geq 0,\ \alpha_j \geq 0$, and estimating the class densities according to

$$\hat{p}(\mathbf{x}) = \frac{1}{\Omega}\sum_{i=1}^{N_1} \alpha_i W_{\sigma^2}(\mathbf{x}, \mathbf{x}_i), \quad \hat{q}(\mathbf{x}) = \frac{1}{\Omega}\sum_{j=1}^{N_2} \alpha_j W_{\sigma^2}(\mathbf{x}, \mathbf{x}_j). \tag{56}$$

The weights, and hence $\hat{p}(\mathbf{x})$ and $\hat{q}(\mathbf{x})$, are learned by enforcing a regularization criterion

$$\min_{\alpha_i,\, \alpha_j} \ \Omega^2 \int [\hat{p}(\mathbf{x}) - \hat{q}(\mathbf{x})]^2 d\mathbf{x}, \tag{57}$$

subject to the classification constraints,

$$\begin{aligned}\hat{p}(\mathbf{x}_i) - \hat{q}(\mathbf{x}_i) + b \geq +\frac{1}{\Omega}, \quad \forall \mathbf{x}_i \in \omega_1, \\ \hat{p}(\mathbf{x}_j) - \hat{q}(\mathbf{x}_j) + b \leq -\frac{1}{\Omega}, \quad \forall \mathbf{x}_j \in \omega_2.\end{aligned} \tag{58}$$

7. Conclusions

We have shown that Parzen window-based estimators for the quadratic information measures are equivalent to Mercer kernel feature space measures, which can be expressed as functions of *mean values* in the Mercer kernel feature space. The Mercer kernel and the Parzen window are shown to be equivalent. This implies that Parzen window size selection procedures known from statistics can be incorporated into Mercer kernel-based methods in order to determine a proper data-driven Mercer kernel size. This also means that the problems associated with applying the Parzen window technique on high dimensional data sets are equally problematic for some Mercer kernel-based methods. Note that this equivalence *can not* be obtained using Parzen window-based estimators for the Shannon measures.

We have proposed a new classification rule based on the ISE measure, combined with Parzen windowing. The resulting classifier was shown to have a dual interpretation as a hyperplane classifier in a Mercer kernel feature space. This observation led us to propose weighted mean vectors in the kernel space in order to be able to regularize the classifier hyperplane by minimizing the squared norm of the weight vector. This approach was related to weighted Parzen window estimators, where the weights were optimized based on the ISE between the density estimators. We showed that this approach could be related to the support vector machine.

In future work, the theory presented here should be evaluated and analyzed in terms of classification experiments.

Some other issues which should be explored are the following. In our information theoretic framework, we estimate the densities non-parametrically for each class. The theory presented in this paper could easily be extended such that a different Parzen window size can be determined for each class. Recall that in the SVM theory, a single kernel size is used. Perhaps performance could be improved by learning different kernel sizes for each class.

Also, in our information theoretic framework, the SVM weights, which are related to weighted Parzen window density estimators, are determined by minimizing the ISE between the class densities. Perhaps some other criteria could be used to learn proper weights. Preferably, such alternative methods should be easier to implement than the SVM optimization. In particular, we will investigate whether the Cauchy-Schwarz pdf divergence measure could be more advantageous in some respect than the the integrated squared error criterion.

Acknowledgments

This work was partially supported by NSF grant ECS-0300340. Deniz Erdogmus was with the Computational NeuroEngineering Laboratory during this work. Robert Jenssen would like to express gratitude to the University of Tromsø, for granting a research scholarship for the academic year 2002/2003, and a two-month research scholarship in the spring of 2004, for visiting the Computational NeuroEngineering Laboratory at the University of Florida.

8. Appendix: Using non-Gaussian Mercer kernels

In this Appendix, we will examine Parzen window-based estimator of $\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}$, using non-Gaussian Mercer kernels.

First, note that

$$\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x} = E_p\{q(\mathbf{x})\}, \tag{59}$$

where $E_p\{\cdot\}$ denotes expectation with respect to the density $p(\mathbf{x})$.

The expectation operator may be *approximated* based on the available samples, as follows

$$E_p\{q(\mathbf{x})\} \approx \frac{1}{N_1}\sum_{i=1}^{N_1} q(\mathbf{x}_i). \tag{60}$$

Assume now that

$$\hat{q}(\mathbf{x}) = \frac{1}{N_2}\sum_{j=1}^{N_2} k(\mathbf{x}, \mathbf{x}_j), \tag{61}$$

where $k(\mathbf{x}, \mathbf{x}_j)$ is a non-Gaussian Mercer/Parzen kernel. Eq. (59) can now be approximated by

$$\begin{aligned}\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x} &\approx \frac{1}{N_1}\sum_{i=1}^{N_1}\hat{q}(\mathbf{x}_i) \\ &= \frac{1}{N_1}\sum_{i=1}^{N_1}\frac{1}{N_2}\sum_{j=1}^{N_2} k(\mathbf{x}_i, \mathbf{x}_j) \\ &= \frac{1}{N_1 N_2}\sum_{i,j=1}^{N_1,N_2} k(\mathbf{x}_i, \mathbf{x}_j).\end{aligned} \tag{62}$$

Hence, the same result is obtained as in the case where Gaussian Parzen kernels were used. However, in this case, it required an additional approximation with regard to the expectation operator. Of course, the same reasoning can be used to approximate the quantities $\int p^2(\mathbf{x})d\mathbf{x}$ and $\int q^2(\mathbf{x})d\mathbf{x}$.

References

G. A. Babich and O. I. Camps. Weighted Parzen Windows for Pattern Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):567–570, 1996.

C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Knowledge Discovery and Data Mining*, 2(2):121–167, 1998.

C. Cortez and V. N. Vapnik. Support Vector Networks. *Machine Learning*, 20:273–297, 1995.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, 2000.

L. Devroye. On Random Variate Generation when only Moments or Fourier Coefficients are Known. *Mathematics and Computers in Simulation*, 31:71–89, 1989.

L. Devroye and G. Lugosi. *Combinatorial Methods in Density Estimation*. Springer-Verlag, New York, 2001.

I. S. Dhillon, Y. Guan, and B. Kulis. Kernel K-means, Spectral Clustering and Normalized Cuts. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 551–556, Seattle, USA, August 22-25, 2004.

D. Erdogmus, K. E. Hild, and J. C. Principe. Blind Source Separation using Renyi's α-Marginal Entropies. *Neurocomputing*, 49:25–38, 2002.

D. Erdogmus, K. E. Hild, J. C. Principe, M. Lazaro, and I. Santamaria. Adaptive Blind Deconvolution of Linear Channels using Renyi's Entropy with Parzen Window Estimation. *IEEE Transactions on Signal Processing*, 52(6):1489–1498, 2004a.

D. Erdogmus, K. E. Hild, Y. N. Rao, and J. C. Principe. Minimax Mutual Information Approach for Independent Component Analysis. *Neural Computation*, 16:1235–1252, 2004b.

D. Erdogmus and J. C. Principe. An Error-Entropy Minimization Algorithm for Supervised Training of Nonlinear Adaptive Systems. *IEEE Transactions on Signal Processing*, 50(7): 1780–1786, 2002a.

D. Erdogmus and J. C. Principe. Generalized Information Potential Criterion for Adaptive System Training. *IEEE Transactions on Neural Networks*, 13(5):1035–1044, 2002b.

D. Erdogmus and J. C. Principe. Convergence Properties and Data Efficiency of the Minimum Error-Entropy Criterion in Adaline Training. *IEEE Transactions on Signal Processing*, 51(7):1966–1978, 2003.

J. H. Friedman. On Bias, Variance, 0/1 Loss, and the Curse-Of-Dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997.

M. Girolami. Mercer Kernel-Based Clustering in Feature Space. *IEEE Transactions on Neural Networks*, 13(3):780–784, 2002a.

M. Girolami. Orthogonal Series Density Estimation and the Kernel Eigenvalue Problem. *Neural Computation*, 14(3):669–688, 2002b.

T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The Entire Regularization Path for the Support Vector Machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

S. Haykin, editor. *Unsupervised Adaptive Filtering: Volume 1, Blind Source Separation*. John Wiley & Sons, New York, 2000.

R. Jenssen, J. C. Principe, and T. Eltoft. Information Cut and Information Forces for Clustering. In *Proceedings of IEEE International Workshop on Neural Networks for Signal Processing*, pages 459–468, Toulouse, France, September 17-19, 2003.

M. Lazaro, I. Santamaria, D. Erdogmus, K. E. Hild II, C. Pantaleon, and J. C. Principe. Stochastic Blind Equalization Based on PDF Fitting using Parzen Estimator. *IEEE Transactions on Signal Processing*, 53(2):696–704, 2005.

Y. A. LeCun, L. D. Jackel, L. Bottou, A. Brunot, C. Cortes, J.S. Denker, H. Drucker, I. Guyon, U. A. Müller, E. Säckinger, P. Y. Simard, and V. N. Vapnik. Learning Algorithms for Classification: A Comparison on Handwritten Digit Reconstruction. *Neural Networks*, pages 261–276, 1995.

M. Di Marzio and C. C. Taylor. Kernel Density Classification and Boosting: An L_2 Analysis. *Statistics and Computing (to appear)*, 2004.

J. Mercer. Functions of Positive and Negative Type and their Connection with the Theory of Integral Equations. *Philos. Trans. Roy. Soc. London*, A:415–446, 1909.

S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller. Fisher Discriminant Analysis with Kernels. In *Proceedings of IEEE International Workshop on Neural Networks for Signal Processing*, pages 41–48, Madison, USA, August 23-25, 1999.

K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An Introduction to Kernel-Based Learning Algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

K. R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. N. Vapnik. Predicting Time Series with Support Vector Machines. In *Proceedings of International Conference on Artificial Neural Networks - Lecture Notes in Computer Science, Springer-Verlag*, volume 1327, pages 999–1004, Berlin, 1997.

E. Parzen. On the Estimation of a Probability Density Function and the Mode. *The Annals of Mathematical Statistics*, 32:1065–1076, 1962.

F. Perez-Cruz and O. Bousquet. Kernel Methods and Their Potential Use in Signal Processing. *IEEE Signal Processing Magazine*, pages 57–65, May 2004.

J. Principe, D. Xu, and J. Fisher. Information Theoretic Learning. In *Unsupervised Adaptive Filtering*, volume I, S. Haykin (Ed.), John Wiley & Sons, New York, 2000a. Chapter 7.

J. C. Principe, D. Xu, Q. Zhao, and J. W. Fisher. Learning From Examples with Information Theoretic Criteria. *Journal of VLSI Signal Processing*, 26(1):61–77, 2000b.

A. Renyi. On Measures of Entropy and Information. *Selected Papers of Alfred Renyi, Akademiai Kiado, Budapest*, 2:565–580, 1976a.

A. Renyi. Some Fundamental Questions of Information Theory. *Selected Papers of Alfred Renyi, Akademiai Kiado, Budapest*, 2:526–552, 1976b.

V. Roth and V. Steinhage. Nonlinear Discriminant Aanalysis using Kernel Functions. In *Advances in Neural Information Processing Systems 12,*, pages 568–574, MIT Press, Cambridge, 2000.

I. Santamaria, D. Erdogmus, and J. C. Principe. Entropy Minimization for Supervised Digital Communications Channel Equalization. *IEEE Transactions on Signal Processing*, 50(5):1184–1192, 2002.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, 2002.

B. Schölkopf, A. J. Smola, and K. R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10:1299–1319, 1998.

D. W. Scott. *Multivariate Density Estimation*. John Wiley & Sons, New York, 1992.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, 1986.

V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

P. Viola and W. M. Wells. Alignment by Maximization of Mutual Information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

P. A. Viola, N. N. Schraudolph, and T. J. Sejnowski. Empirical Entropy Manipulation for Real-World Problems. In *Advances in Neural Information Processing Systems, 8*, pages 851–857, MIT Press, Cambridge, 1995.

M. P. Wand and M. C. Jones. *Kernel Smooting*. Chapman and Hall, London, 1995.

D. Xu. *Energy, Entropy and Information Potential for Neural Computation*. PhD thesis, University of Florida, Gainesville, FL, USA, 1999.

A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K. R. Müller. Engineering Support Vector Machine Kernels that Recognize Translation Invariant Sites in DNA. *Bioinformatics*, 16:906–914, 2000.

Chapter 6

Paper 4:
Independent Component Analysis for Texture Segmentation

PERGAMON

Pattern Recognition 36 (2003) 2301–2315

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Independent component analysis for texture segmentation

R. Jenssen*, T. Eltoft

Department of Physics, University of Tromsø, N-9037 Tromsø, Norway

Received 13 September 2002; received in revised form 28 March 2003; accepted 28 March 2003

Abstract

Independent component analysis (ICA) of textured images is presented as a computational technique for creating a new data dependent filter bank for use in texture segmentation. We show that the ICA filters are able to capture the inherent properties of textured images. The new filters are similar to Gabor filters, but seem to be richer in the sense that their frequency responses may be more complex. These properties enable us to use the ICA filter bank to create energy features for effective texture segmentation. Our experiments using multi-textured images show that the ICA filter bank yields similar or better segmentation results than the Gabor filter bank.

Keywords: Independent component analysis; Image model; Data dependent filter bank; Texture segmentation; Energy features

1. Introduction

Automatic segmentation of multi-textured images is an important area of image processing which has received considerable attention during the last decade. In this paper we introduce *independent component analysis* (ICA) [1–4] of textured images as a computational technique for creating a new *data dependent* filter bank for use in texture segmentation.

Independent component analysis is a generative model for observed multivariate data, which are assumed to be mixtures of some unknown latent variables. Originally, it emerged as a means to solve the problem of blind source separation (BSS) [3,5,6]. BSS refers to the problem of finding the original source signals from available mixtures, without any prior knowledge of the number of sources or the mixing mechanisms. In the linear case, the observed signals $\mathbf{x}(t)$ can be expressed as

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), \tag{1}$$

where $\mathbf{A}$ is called the mixing matrix and $\mathbf{s}(t)$ is the source signals.

Based on the assumption that the source signals are mutually statistically independent, the solution is obtained in an unsupervised learning process, which finds the de-mixing matrix $\mathbf{W}$, such that $\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t)$ is an estimate of $\mathbf{s}(t)$. The components of $\mathbf{y}(t)$ are required to be as mutually independent as possible.

The model in Eq. (1) is referred to as the basic ICA model [4]. Thus, in blind source separation it is the source signals which are the "independent components". This model has in the recent years been widely applied in diverse areas like biomedical imaging [7–10], telecommunications [11,12], time series prediction [13–15], audio separation [16] and seismic monitoring [17]. See e.g. Refs. [18–20] for more references on ICA applications.

ICA has also been proposed as a generic statistical model for images [21–25]. In this case an image $\mathbf{x}$ is modeled as

$$\mathbf{x} = \sum_{i=1}^{N} s_i \mathbf{a}_i, \tag{2}$$

where $\mathbf{a}_i, i = 1,\ldots,N$, are referred to as ICA image basis functions, and $s_i, i = 1,\ldots,N$, are statistically independent weighting components. It also turns out that the s_i-components exhibit so-called sparseness, i.e. only a few of the weighting coefficients will have a value significantly deviating from zero. This property is exploited when the

* Corresponding author. Tel.: +47-776-45184; fax: +47-776-45580.

E-mail address: robertj@phys.uit.no (R. Jenssen).

0031-3203/03/$30.00
doi:10.1016/S0031-3203(03)00131-6

model is used in threshold-based image denoising (sparse code shrinkage) [26,27]. Eq. (2) was in Ref. [21] originally motivated as a model for the neurons in the primary sensory areas of the brain.

In this paper we propose to use the ICA image model of Eq. (2) to generate a *data dependent filter bank for texture segmentation*. The filter bank consists of the ICA basis images, learned from textured images. We show that these basis images are able to capture the inherent structure of the texture, and hence enable us to create features for effective texture segmentation of composite input textures.

Traditionally, texture segmentation techniques are classified as *statistical methods*, *model based methods* or *filtering methods*.

The statistical methods are often based on the co-occurrence matrix [28], which gives information about the second-order statistics of a textured image. Texture features which can be extracted from the co-occurrence matrix, are based on parameters quantifying e.g. energy, entropy, contrast, homogeneity or correlation [29–31].

In model based methods a textured image is modeled as a probability model, and the coefficients of the model are used to characterize the images [32]. These models include Gaussian Markov random fields [33], Gibbs random fields [34] and Wold models [35,36].

Our ICA based texture segmentation method belongs to the filtering methods, also called signal processing methods [37], which include a whole family of algorithms. This approach also found motivation in psycho-physiology. Studies by e.g. Campbell and Robson [38] and De Valois et al. [39] suggested that the brain performs a multi-channel frequency and orientation analysis of the visual image formed on the retina [37]. The basic idea in the filtering methods is that a composite textured image is filtered through a bank of filters, and features appropriate for texture segmentation are generated based on the filter outputs [40].

Laws' texture energy filters [41,42] represent an early example of this approach. Law's generated 2-D filter kernels designed to detect "spots", "edges" and "rings". The filters are convolved with a composite textured image to produce a set of texture energy images. Each pixel in the input image is hence represented by a vector of energy features. Subsequent methods often follow this outline, but using different filter banks. For example, Coggins and Jain [43] suggested using filters characterized in the spatial frequency domain as ring filters with widths between one and two octaves, and with center frequencies one octave apart. The choice of filter bandwidth and center frequency separation were motivated from experiments showing that the frequency bandwidth of simple cells in the striate cortex is about one octave [44]. Wedge-shaped orientation channels were also implemented, with directions along the horizontal, vertical and the two diagonals.

Marcelja [45] and Daugman [46] showed that 2-D Gabor filters are good models of the receptive fields in the striate cortex. In the last couple of decades Gabor filters have been widely used in texture segmentation [47–51]. Gabor filters can be interpreted as oriented band-pass filters [49].

Jain and Farrokhnia [49] used a dyadic even-symmetric Gabor filter bank to generate energy features for texture segmentation, and reported successful results. Their filters are oriented along the horizontal, vertical and the two diagonals, and the number of center frequencies, i.e. the size of the filter bank, varies according to the size of the textured image. We will use Jain and Farrokhnia's method for comparison with ICA based texture segmentation.

Wavelet and wavelet packets methods [52–57] of various sub-band decompositions have also been successfully applied in texture analysis. With an appropriately chosen basis set, the 2-D wavelet transform offers spatial frequency and orientation selectivity, and texture features at the resolution levels containing most of the energy activity can be obtained [29,52]. The Gabor filters form an approximate wavelet basis, with the Gabor function as the wavelet [49].

The ICA based approach is different from existing filtering methods in that it produces a data dependent filter bank. The filters are obtained from training data using an unsupervised algorithm, and the user is hence required to provide training data containing the appropriate structure. Such data may be found in available image or texture libraries.

In the next section we present in some detail independent component analysis as a generic statistical model for images. Next, in Section 3 we specifically discuss the properties of the ICA filter bank obtained from a training set of texture data. We also briefly review the Gabor filter theory. In Section 4 we explain how the ICA filter bank method can be used for feature generation, and in Section 5 we present some texture segmentation experiments where we compare the performance of the ICA filter bank and the Gabor filter bank. In Section 6, we make some concluding remarks.

Finally, for the convenience of the reader unfamiliar with ICA, we include in Appendix A a brief review of some basic ICA theory.

2. Independent component analysis of images

Independent component analysis of images produces a set of basis functions [21,25,24]. Hence, as noted from Eq. (2), an image may be expressed as a weighted sum of these basis functions, with weighting components $s_i, i = 1,\ldots,N$. The basic idea in the ICA image model is to construct basis functions which for a given image result in weighting components which are mutually statistically independent. Thus, the s_i's in Eq. (2) are the "independent components". The image model described here is illustrated in Fig. 1.

In the sequel, we represent an image as a column vector $\mathbf{x} = [x_1,\ldots,x_M]^T$ by re-shaping the image matrix row-by-row into a single column. In this way we avoid using four-dimensional matrices in the ICA filtering, and

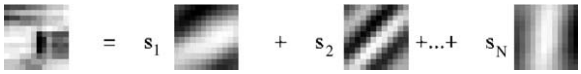


Fig. 1. The ICA model. An image is modeled as a weighted summation of basis images. The weights are the "independent components".

an image may thus be represented using the standard ICA methodology. Mathematically, we express the model as,

$$\mathbf{x} = \sum_{i=1}^{N} s_i \mathbf{a}_i = \mathbf{A}\mathbf{s}, \tag{3}$$

where the basis functions $\mathbf{a}_i, i = 1,\ldots,N$, are the columns of the $(M \times N)$ matrix $\mathbf{A}$, and $\mathbf{s} = [s_1,\ldots,s_N]^{\mathrm{T}}$.

In order to learn the mixing matrix, $\mathbf{A}$, we attempt to find a linear transformation $\mathbf{W}$ of the data $\mathbf{x}$, which yields a vector $\mathbf{y}$, estimating $\mathbf{s}$, whose components are as statistically independent as possible. Accordingly

$$\mathbf{y} = \mathbf{W}\mathbf{x}. \tag{4}$$

If $N = M$, $\mathbf{W}$ is the inverse of $\mathbf{A}$, i.e. $\mathbf{W} = \mathbf{A}^{-1}$. There exists no analytical solution to this problem.

In some cases it is expedient to choose $N < M$, i.e. the dimension of $\mathbf{y}$ is less than the dimension of $\mathbf{x}$. Dimensionality reduction can be accomplished by principal component analysis (PCA). PCA may be considered a preprocessing step towards statistical independence, since it removes second order correlations in the training data. This is often called whitening. The PCA preprocessing and dimensionality reduction is discussed in Appendix A.

Since the weighting components in the ICA model are mutually statistically independent, one might say that ICA attempts to produce a code $\mathbf{y}$ of $\mathbf{x}$, where the joint probability density $f(\mathbf{y})$ of the components of $\mathbf{y}$ factorize as

$$f(\mathbf{y}) = \Pi_i f(y_i). \tag{5}$$

The functions $f(y_i), i = 1,\ldots,N$ are the estimates of the probability densities of the individual s_i-components. This is sometimes referred to as a factorial code [21].

The matrix $\mathbf{W}$ is determined using a statistical approach. Training data is presented to an unsupervised algorithm optimizing statistical independence of the components of $\mathbf{s}$ by iteratively adjusting an initially randomly chosen matrix $\mathbf{W}$. The training data are considered realizations of random vector $\mathbf{X}$. There exist several algorithms performing ICA [58–63]. We have used Hyvärinens FastICA algorithm [62,63], which is briefly discussed in Appendix A.

The ICA basis functions are data dependent in the sense that they are learned from the training data at hand, and they will be different for different training data. The basis functions can be considered as image building blocks, capturing the inherent features of the training data. Several authors have applied ICA in order to reveal the features of natural images [21,24–26], that is, images void of any man made structures. In that case, the training data must be generated from images considered to belong to the class of natural images. When modeling so-called man-made images, images of buildings, cars, etc. are used to generate training data, as in Ref. [26]. We are interested in ICA basis functions which can capture the inherent properties of texture, thus in this paper the training data are generated from textured images.

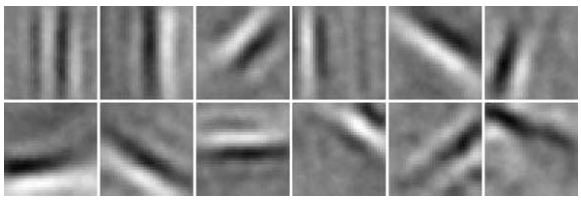


Fig. 2. Independent basis functions of size (32×32) learned from natural images.

2.1. Independent component analysis of natural images

In recent years it has been shown that ICA applied to natural images generates a very interesting set of basis functions [21,23–26]. Barlow [64,65] studied coding principles which could predict the formation of localized, oriented receptive fields in the visual cortex [66]. He suggested that the cortical feature detectors might be the end result of a redundancy reduction process, in which the activation of each feature detector should be as statistically independent as possible from the activation of the others. This implies a factorial code. Bell and Sejnowski [21] reconsidered the theories of Barlow and developed an information maximization network, which performs ICA, and achieved results as predicted by Barlow's theory. They found ICA basis functions which were localized in space, e.g. a single edge at a certain location, and of a multiscale nature [21]. Their findings were quite close to those obtained by the related sparseness maximization network of Olshausen and Field [67].

Similar findings have also been reported in e.g. [23,26,25]. The ICA basis functions have been shown to be localized Gabor-like functions of various orientations, spatial frequencies and phases, and to bear a striking resemblance to the receptive fields of neurons in the primary visual cortex [25]. Fig. 2 shows 6 examples of ICA basis functions created using the same natural images as were used in Refs. [23,26]. The characteristics mentioned above, can be clearly noted.

In Ref. [26] the ICA representation of man-made scenes was studied. These basis functions were found to have continuous lines and edges, and appear quite different from those obtained from natural images.

3. The ICA representation of texture

In this section we discuss in detail how ICA can be used to obtain a data dependent filter bank representation of textured images. For comparison we also briefly review the

Gabor filter bank representation, and give some comparative remarks.

3.1. Learning methodology

In order to construct ICA basis functions of size $(m \times m)$, the columns $\mathbf{a}_i, i = 1, \ldots, N$, of $\mathbf{A}$ in Eq. (3) must be of size $(M \times 1)$, where $M = m^2$. This will also be the case for the training data $\mathbf{x}$. In our case, we have available a set of 20 textured images, and each $\mathbf{x}$ is generated by extracting an $(m \times m)$ image patch from one of the training textures, and representing it as a column vector. We generate a large number of training vectors (in our case 15 000), ensuring that each training texture provides an equal amount of training vectors.

Note that the representation of $(m \times m)$ image patches as column vectors destroys part of the two-dimensional correlations that help characterize texture. We try to compensate for this by repeatedly rotating the textured training images when extracting the image patches. Therefore, before the selection of training patches, a texture image is rotated a random angle θ, where $\theta \in [0^\circ, 180^\circ]$. Each texture image is rotated several angles to give a number of patches from different orientations, and every sample vector is taken from a random location in the rotated image. Also, to ensure that no texture dominates the basis set, each training texture image is normalized and made zero mean prior to the sampling. This results in a zero ensemble mean of the training vectors.

In addition, every training vector $\mathbf{x}$ is subtracted its individual local mean value. If this is not done, one of the resulting basis functions will represent the mean intensity value of the training data, which is of little value in the filter bank context. Subtraction of the local mean value has the effect that one of the PCA components gets the variance zero [22]. Therefore, without loosing any information, we may reduce the dimension in PCA space by one, and hence, the maximum number of basis functions we need to estimate is $N = M - 1$.

In summary, the generation of training data includes the following computations:

- Normalize training texture images and make them zero mean
- Sampling loop: *for* $i = 1$ *to number-of-images*
  - Rotation loop: *for* $j = 1$ *to number-of-directions*
    - Rotate training texture number i a random angle θ_j
    - Select the given number of $(m \times m)$ sized patches form random locations
    - Subtract from each patch its mean value
    - Represent samples as columns, and store in a matrix consisting of training vectors

The computations described above result in an ensemble of training vectors which are presented to the FastICA algorithm, after additional preprocessing by PCA. After convergence of the algorithm, matrix $\mathbf{W}$ has been learned from the available training data. Matrix $\mathbf{A}$ is obtained by finding the inverse (or pseudo inverse) of $\mathbf{W}$. As a last step, the columns $\mathbf{a}_i, i = 1, \ldots, N$, are re-shaped into size $(m \times m)$ to constitute the basis functions. These are now the impulse responses of our filter bank.

It is clear that it is computationally demanding to learn ICA basis functions even of a relatively modest size. For example, $m = 12$ requires the ICA algorithm to handle training data of size (144×1), leading to a maximum of 143 basis functions. For $m = 256$ the data vector is of size $(65\,536 \times 1)$ leading to maximum 65 535 basis functions. The latter is clearly computationally prohibitive. Therefore, in our segmentation experiments we limit the size of the basis functions to (12×12), and we also apply extensive dimension reduction as provided by PCA.

3.2. Characteristics of ICA filter bank

All the textures we use for training the ICA basis functions are taken from the Brodatz album [68]. Fig. 3 (a) shows the 20 textures, they are all of size (640×640). Fig. 3(b) shows an example of (12×12) ICA basis functions learned by the method outlined above. The dimension are reduced by PCA, resulting in a total of 50 functions. The basis functions are shown in decreasing order of their dominant frequency component.

We observe that a majority of the functions are localized in spatial frequency and orientation. That is, they seem to exhibit a kind of sinusoidal wave structure of a specific spatial frequency and to be oriented in a specific direction. Fig. 4(a) shows a scatter plot of the dominating frequency component versus the orientation (in degrees) for the filters of Fig. 3. The dominating frequency components range from approximately 0.05 Hz to about 0.38 Hz. The orientations of the functions vary significantly, covering a number of angles in the range 0–180°. We note that the basis functions are fairly uniformly spread in frequency and orientation.

We also observe that some of the basis functions appear to have a checkerboard-like intensity variation. See e.g. functions number 13, 16, 17, 18 or 19 of Fig. 3, where function number 1 is to the top left of Fig. 3 and function number 50 to the bottom right, read row-wise. This particular attribute obviously occurs because several of the training textures exhibit a checkerboard-like structure.

Fig. 4 shows examples of the frequency domain representation of ICA basis functions number 9 (b) and number 13 (c) of Fig. 3. We observe that function number 9 has a single frequency component, oriented approximately 45° from the horizontal direction. The single dominating frequency component is about 0.2 Hz. The frequency contents of function number 13 is significantly different. In this case the ICA basis function seems to have several frequency components, oriented in different directions. This gives rise to the kind of checkerboard-like structure we observe in the spatial domain.

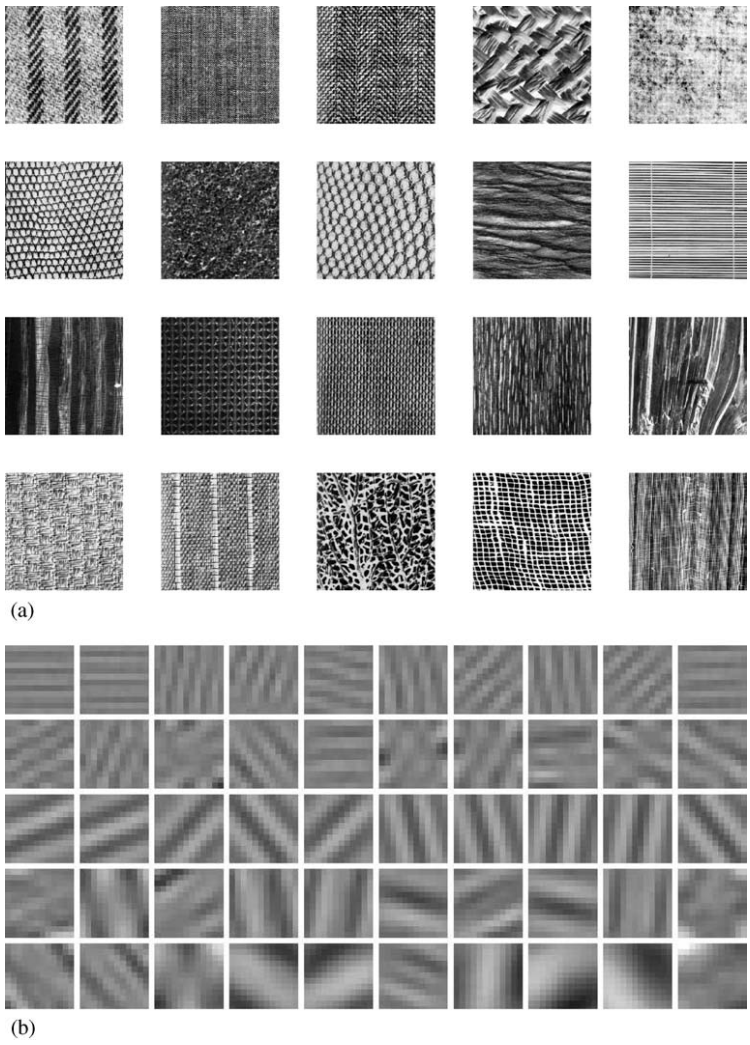


Fig. 3. Training textures and resulting ICA basis functions: (a) Training textures from the Brodatz album, from top left to bottom right: $D11$, $D16$, $D17$, $D18$, $D19$, $D22$, $D29$, $D35$, $D37$, $D49$, $D50$, $D52$, $D53$, $D68$, $D72$, $D82$, $D85$, $D87$, $D103$, and $D105$; (b) example of (12×12) ICA basis functions for texture.

We note that the region of support of the impulse responses of the ICA basis functions in Fig. 3 are not localized to a certain region in space, but they extend throughout the whole patch size. Since textures typically consist of repeating patterns covering the whole image, this can be expected when textures are used as training data. This is in contradiction to ICA basis functions obtained from natural images, which typically have a region of support which is localized in space, like e.g. an edge.

Next, we address the question of how well the information contents of an image is preserved when the number of ICA basis functions is reduced. Consider retaining only l basis functions of a (8×8) sized learning set, where l is less than or equal to 63. Denote the mixing matrix $\mathbf{A}_l$ and the de-mixing matrix $\mathbf{W}_l$. We project a data vector $\mathbf{x}$ into the l-dimensional ICA space by the transformation

$$\mathbf{y}_l = \mathbf{W}_l \mathbf{x}, \tag{6}$$

and reconstruct an estimate $\hat{\mathbf{x}}_l$ of $\mathbf{x}$ as

$$\hat{\mathbf{x}}_l = \mathbf{A}_l \mathbf{y}_l. \tag{7}$$

How close is $\hat{\mathbf{x}}_l$ to $\mathbf{x}$? Fig. 5(b)–(d) shows the result of estimating $\mathbf{x}$ using $l = 10$, 30 and 63. The original $\mathbf{x}$ is also shown in Fig. 5 (a). We observe that even 10 basis functions maintains the basic structure of the image, yielding a normalized squared error (NSE) of $NSE \approx 0.4$. The reconstruction is of course better for 30 basis functions, with $NSE \approx 0.1$. Using all 63 basis functions yields nearly perfect reconstruction, resulting in an $NSE \approx 0$.

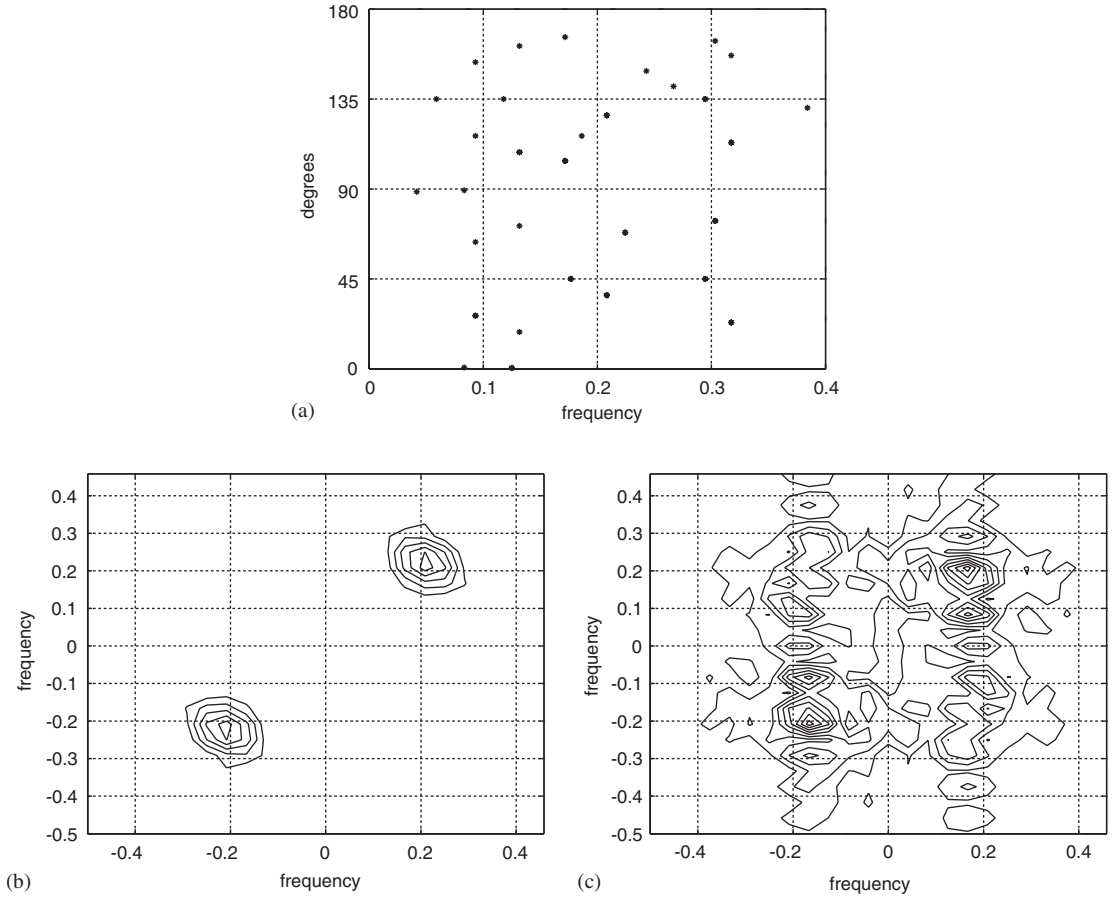


Fig. 4. Characterization of ICA filters: (a) Scatter plot of dominating frequency component versus orientation (in degrees) for the ICA basis functions of Fig. 3. (b,c) Contour plots of frequency domain representation of ICA basis functions number 9 and number 13, respectively.

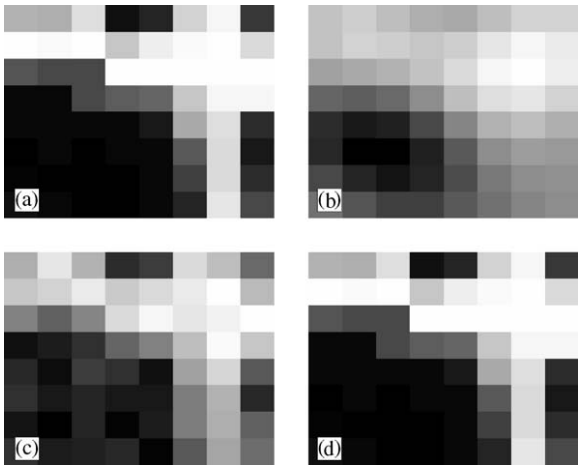


Fig. 5. Reconstruction of original (8×8) image using ICA basis functions. (a) Original image, (b) using 10 basis functions yielding $NSE \approx 0.4$, (c) using 30 basis functions, $NSE \approx 0.1$ and (d) using 63 basis functions, $NSE \approx 0$.

The above experiment shows that the basic characteristics of textured images are represented in an ICA basis of reduced dimensions (see Appendix A for details).

3.3. The Gabor filter bank

The impulse response of an even-symmetric Gabor filter is given by

$$h(x, y) = g[q(x, y), w(x, y)] \cos[2\pi f_0 q(x, y)], \tag{8}$$

where

$$g(x, y) = \exp\left\{-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right\}, \tag{9}$$

$$q(x, y) = \cos(\Phi_0)x + \sin(\Phi_0)y, \tag{10}$$

$$w(x, y) = -\sin(\Phi_0)x + \cos(\Phi_0)y. \tag{11}$$

We note that $g(x, y)$ is the Gaussian envelope with space constants σ_x and σ_y along the x and y axes, respectively. The Gaussian envelope is rotated an angle Φ_0 with respect to the

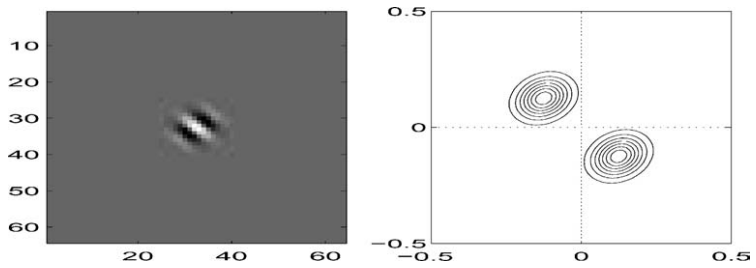


Fig. 6. Example of a Gabor filter in a (64×64) grid. Left: spacial domain, right: frequency domain. Normalized frequency is $f_0 = \sqrt{2}/2^3$, and orientation 135°.

x-axis (the 0° orientation), and multiplied by a sinusoidal plane wave of spatial frequency f_0, yielding $h(x, y)$.

The frequency domain representation of Eq. (8) is given by

$$\frac{H(u,v)}{2\pi\sigma_x\sigma_y} = \exp\left[-\frac{(q(u,v)-f_0)^2}{2\sigma_u^2} - \frac{w^2(u,v)}{2\sigma_v^2}\right] + \exp\left[-\frac{(q(u,v)+f_0)^2}{2\sigma_u^2} - \frac{w^2(u,v)}{2\sigma_v^2}\right], \quad (12)$$

where $\sigma_u = 1/(2\pi\sigma_x)$ and $\sigma_v = 1/(2\pi\sigma_y)$. Jain and Farrokhnia [49] performed filtering by multiplying $H(u,v)$ of Eq. (12) with the 2-D Fourier transform of the composite textured image.

For the Gabor filter defined by Eq. (12), the half-peak magnitude frequency bandwidth, B_r, and orientation bandwidth, B_θ, are given by [49]

$$B_r = \log_2\left(\frac{f_0 + \sqrt{2\ln 2}\sigma_u}{f_0 - \sqrt{2\ln 2}\sigma_u}\right), \quad (13)$$

$$B_\theta = 2\tan^{-1}\left(\frac{\sqrt{2\ln 2}\sigma_v}{f_0}\right), \quad (14)$$

where B_r is in octaves and B_θ is in degrees. B_r is often chosen to be one octave, while the orientation bandwidth is chosen to be 45°. Thus, for any given f_0, σ_u and σ_v are given by

$$\sigma_u = \frac{f_0}{3\sqrt{2\ln 2}}, \quad (15)$$

$$\sigma_v = \frac{\tan 22.5^\circ f_0}{\sqrt{2\ln 2}}, \quad (16)$$

respectively.

The impulse response and frequency content of a typical Gabor filter is shown in Fig. 6. Note that the region of support of the impulse response is precisely localized in space, and the frequency response is correspondingly localized in frequency. The size of the region of support of the impulse response is determined by the width of the Gaussian envelope.

For an image of width N_c pixels, the following normalized discrete radial center frequencies, f_0, have been suggested for a bank of Gabor filters [40],

$$\frac{1\sqrt{2}}{N_c}, \frac{2\sqrt{2}}{N_c}, \frac{4\sqrt{2}}{N_c}, \ldots, \frac{\frac{1}{4}N_c\sqrt{2}}{N_c}. \quad (17)$$

The above frequency choice ensures that the pass-band of the filter with the highest frequency falls below the Nyquist frequency. For some textures the lowest radial frequencies of Eq. (17) are not very useful because they capture features too coarse to represent textural variations in an image [57], and are therefore often omitted. Hence, for a (256×256) image the following radial frequencies are used:

$$\frac{\sqrt{2}}{2^6}, \frac{\sqrt{2}}{2^5}, \frac{\sqrt{2}}{2^4}, \frac{\sqrt{2}}{2^3}, \frac{\sqrt{2}}{2^2}. \quad (18)$$

The filters are oriented along directions $0^\circ, 45^\circ, 90^\circ$ and 135°.

3.4. Comparison between ICA and Gabor filters

In this subsection we point out some apparent differences between the traditional Gabor filters and our ICA filters.

The Gabor transform is a special case of the window Fourier transform (short-time Fourier transform) [37], where the window function is Gaussian. The window Fourier transform is a local analysis of the frequency content of an image. The analysis is limited by the effective width of the filter in the spatial domain or equivalently by its bandwidth in the frequency domain. Hence, from Eqs. (15) and (16) the size of the region of support of a Gabor filter is effectively inversely related to its center frequency. Gabor filters have the important property of having optimal joint localization, or resolution, in both the spatial and the spatial frequency domain [49].

In contrast, the ICA filters are finite support filters whose size is determined by the size of the image patches selected. They are data dependent in that they have been constructed from the available training data. This allows the ICA filters to vary in shape and frequency contents as can noted from Fig. 3. The Gabor filters are characterized by their orientation and their center frequency. The ICA filters shown in Fig. 3 are richer, they may be frequency and orientation selective like Gabor filters (see e.g. filter 9), but they may also

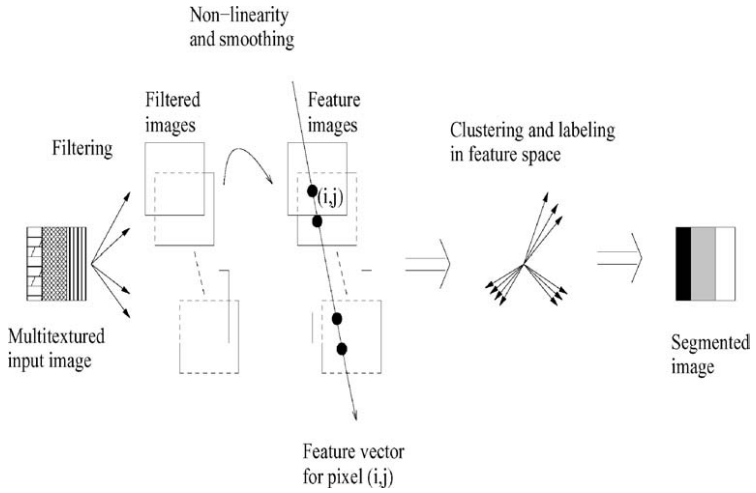


Fig. 7. Experimental setup for segmenting a multi-textured input image.

realize multiple pass-bands of multiple orientations (see e.g. filter 13).

4. Generation of features for texture segmentation

To be able to automatically segment a multi-textured image into its distinct segments, it is necessary to generate representatives of each pixel which can capture the inherent nature of the local pattern surrounding it. These representatives usually form vector valued functions of the image pixels, denoted as feature vectors.

If the textures have sufficiently different structure and the feature vectors are well designed, they will form compact clusters in feature space, each cluster corresponding to a distinct texture region in the multi-textured input image. The goal is to reveal the clusters in feature space by the use of some clustering algorithm and subsequently assign pixels to their appropriate texture class.

The basic idea in filter based texture segmentation has already been briefly described previously. We filter the original image through the bank of filters, each with specific frequency and orientation characteristics. If the bank consists of N filters, the result is N filtered images of the same size as the input image.

Each filter responds to specific texture properties. Thus, each filtered image will have high energy in regions corresponding to textures which are tuned to the filter, and low energy corresponding to textures which are not tuned to that filter. Then, the approach taken is to apply a non-linearity, such as squaring, followed by a smoothing operation to the filtered images. The resulting images are called feature images, and the feature vectors are formed by collecting corresponding pixel values from each feature image in a vector. The experimental setup described here is shown in Fig. 7.

4.1. Creating feature images

When creating features based on an energy measure, the objective of the non-linearity and smoothing operations, is to estimate the energy in a local region of the filter output. The most commonly used non-linearities are the magnitude function $|\cdot|$ [41,43,48,69], the squaring function $(\cdot)^2$ [56,53,51], or the rectified sigmoid function $|\tanh(\alpha\cdot)|$ [49]. We use the squaring function.

It is furthermore important to determine a proper size and shape of the smoothing window. If the smoothing window is too small, the estimate of the local energy will be unreliable, while if the window is too large, we will smooth out boundaries between texture regions. A Gaussian window is likely to result in more accurate localization of texture boundaries rather than a rectangular window [49]. Jain and Farrokhnia [49] suggested to let the variance of the Gaussian smoothing window be dependent on the center frequency of the Gabor filter according to $\sigma = 1/2\sqrt{2}f_0$. We find that using $\sigma = 2/\sqrt{3}f_0$ yields better results in our experiments for both the ICA and the Gabor filter banks. Also, the size of the window can be fixed to a preselected value, or determined automatically. We choose the size of the window to be $(l \times l)$, where $l = 2\sigma$. Thus, the Gaussian window width is also different for each filter in the filter bank.

In addition, as a last step, we apply a logarithmic normalizing non-linearity. The combination of squaring and logarithmic normalization was found by Unser and Eden [70] to give very good texture segmentation performance. We use this normalization. Denoting the kth filtered image by $r_k(x, y)$, the kth feature image $e_k(x, y)$ is then given by

$$e_k(x, y) = \log\left(\frac{1}{l^2}\sum_{a,b\in W} r_k^2(x-a, y-b)W(a, b)\right), \quad (19)$$

where W is the Gaussian window centered at pixel coordinates (x, y).

4.2. Filter selection

Filter selection refers to the problem of selecting a subset of the feature generating filters which exhibit high information packing in the sense of having high texture separability performance. The problem is: which filters to choose, and how many.

Referring to Fig. 3, we observe that some of the ICA basis functions look quite similar. These functions will have a high degree of similarity also in their frequency domain representations, and are expected to yield almost identical outputs when used to filter an input textured image. This means that the features based on these filters will provide us with essentially the same information and thus bring redundancy into the feature vectors.

Our first filter selection step is to find the subset of J filters with the smallest degree of overlap in the frequency domain. We simply form the frequency representations, i.e. the transfer functions, F_i and F_j of two filters i and j, and measure their frequency overlap by calculating $|F_i - F_j|$. This is done pairwise for all the filters, and the J filters having mutually the smallest overlap are chosen. All filters are normalized in energy before the selection procedure.

Our next step is to examine the feature images of all the J remaining filters, to obtain a ranking based on whether the features possess variability over the textured image or not. If a feature image does not have variability, the feature does not provide local information. In doing this ranking, we use the F-test proposed in Ref. [71]. We divide each feature image into R equally sized image regions, $\Omega_r, r = 1, \ldots, R$, and compute the residual sum of squares (RSS) in each of the regions. The RSS of feature image k in region Ω_r, is defined by

$$RSS_{kr} = \sum_{i,j \in \Omega_r} [e_k(i,j) - m_{kr}]^2, \tag{20}$$

where $e_k(i,j)$ is the value of feature no. k in pixel (i,j), and m_{kr} is the mean of the feature in the region Ω_r. We form the total RSS_{kT} as the sum

$$RSS_{kT} = \sum_{r=1}^{R} RSS_{kr}. \tag{21}$$

This value is compared to the RSS_k computed over the whole feature image $e_k(x, y)$,

$$RSS_k = \sum_{i,j} [e_k(i,j) - m_k]^2, \tag{22}$$

where m_k is the mean value of the kth feature image. If RSS_{kT} is close to RSS_k, the feature does not provide local information.

The F-test is based on [71]

$$F_k = \frac{(N-R)(RSS_k - RSS_{kT})}{(R-1)RSS_{kT}}, \tag{23}$$

where N is the total number of pixels in each of the regions $\Omega_r, r = 1, \ldots, R$.

When it comes to the number of filters to be used in the filter bank, one possibility is to use those filters yielding values of F_k above a significance-threshold U_T. The threshold is determined from the feature corresponding to the largest F value.

4.3. Clustering in feature space

We use the well-known K-means algorithm [29,37] to cluster the feature vectors. This is a simple algorithm, which also has been widely applied in the texture segmentation context. The implementation, i.e. the initialization of the cluster prototypes, may be done in several ways, and the clustering result is often dependent on which initialization procedure is used. We initialize the prototypes by randomly drawing prototypes from the set of feature vectors. This method has the disadvantage that it may converge to a local minimum in the K-means cost function, which in our case is the sum of squared prototype distance. To avoid this problem, we run the algorithm several times, and choose the best clustering as our final result.

A well known problem with the K-means algorithm is that it requires a priori knowledge of the number of clusters to be formed. For a fully automatic texture segmentation algorithm it is therefore of crucial importance to be able to obtain reliable estimates of this number before running the K-means algorithm. There exists a variety of different cluster validity indicies [72,73] for this purpose. In this paper, the main objective is to introduce ICA generated features, and analyze texture segmentation based on these. We therefore assume that we have a priori knowledge of the number of clusters.

5. Segmentation experiments

In this section, we report results on texture segmentation using the ICA filters. The filters are all learned from the training textures in Fig. 3(a) using (12×12) sized patches. We have previously pointed out the important property of the ICA filters being data dependent. This means that they are sensitive to the training data, and to emphasize this fact, we have tested two basis sets. The first set, shown in Fig. 8(a), is the 20 basis functions with the least frequency overlap, selected among the 50 filters previously shown in Fig. 3(b). They are ordered according to descending F-value (see Section 4.2). Note that both filters number 9 and 13 of Fig. 3, with frequency responses shown in Fig. 4(a) and (b), are among the selected filters (filters number 17 and 13 in Fig. 8(a), respectively). The second set, shown in Fig. 8(b), is generated from the same training textures, but for this set we did not normalize the textured image energy prior to the ICA basis generation.

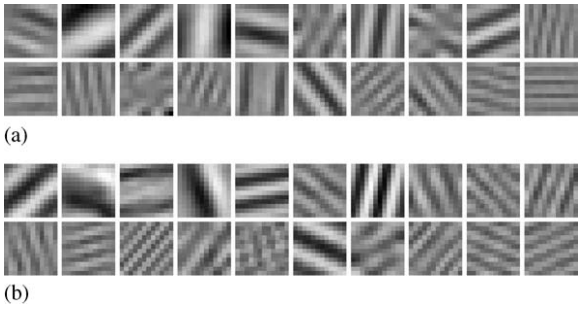


Fig. 8. The 20 most uncorrelated (12×12) ICA functions learned from (a) normalized and (b) non-normalized training textures. The basis functions are ordered from top left to bottom right according to their F-test significance.

The segmentation performance of these two sets are compared to that of a Gabor filter bank consisting 20 filters. The Gabor filters have radial frequencies according to Eq. (18), and are oriented along the directions $0°, 45°, 90°$ and $135°$. In all cases, the width of the Gaussian smoothing windows is given by $\sigma = 2/\sqrt{3} f_0$, where f_0 in the ICA cases is an estimated dominating frequency. The segmentation is performed using the K-means algorithm to cluster the energy feature vectors obtained by filtering the input texture through the respective filter banks.

Two multi-textured input images (256×256) are used. Fig. 9(b) shows the first, which is composed of two different texture patterns. The performance of an ICA filter bank is obviously depending on how many filters are used. This dependency has been depicted in Fig. 9(a), where we have plotted the percentage of mis-classified pixels as function of the number of filters. The dashed curve corresponds to the filters of normalized input textures, and the solid corresponds those of the non-normalized. When extending the filter bank, we include more and more filters following the order they are presented in Figs. 8(a) and (b). We note that the first set almost always performs the best, and an optimum result is obtained with 7 filters, resulting in an error percentage of 2.9%. The second set has its best performance using 20 filters, achieving an error percentage of 3.5%. The segmented images are displayed in Figs. 9(c)–(e), where the white solid curve marks the true boundary between the two texture patterns. We note that both the ICA sets in this case outperform the Gabor filter bank, which achieves a best error rate of 13.8%.

The second multi-textured input image consists of five texture patterns, and is displayed in Fig. 10(b). Fig. 10(a) show how the mis-classification rate changes with the number of filters in the ICA filter banks. In this case the best results are obtained using 13 filters for the first set (normalized input textures), and 11 filters for the second set. The error rates are 2.8% and 3.0%, respectively. Also in this case, the ICA filter banks perform better than the Gabor filter bank, allthough the difference is not as large as in the previous experiment. The best result for the Gabor filterbank is in this case 3.6%. The segmented images are depicted in Figs. 10 (c)–(e).

6. Conclusions

We have presented independent component analysis (ICA) of textured images as a computational technique for creating a new data dependent filter bank for use in texture segmentation.

The ICA filters can be used as basis functions in an image model, in which an image is expressed as a weighted sum of the impulse responses. These basis functions are obtained in an unsupervised learning process, where the criterion for learning is that the weighting components should be as statistically independent as possible. The training data are in this application gathered from a set of textured images. Thus, the ICA filters are data dependent.

We have demonstrated that the ICA filters are able to capture the inherent properties of textured images. A majority of the functions are localized in spatial frequency and orientation. That is, they seem to exhibit a kind of sinusoidal wave structure of a specific spatial frequency and direction. Such filters can be interpreted as oriented band-pass filters. Interestingly, some of the ICA filters have frequency responses with several components, orientated in different directions. The new ICA filter bank is similar to the Gabor filter bank, but it seems to be richer in the sense that some filters have more complex frequency responses.

The above properties allow us to use the ICA filter bank to create energy features for effective texture segmentation. Our experiments using multi-textured images show that the ICA filter bank yields similar or better segmentation results than the Gabor filter bank.

Appendix A. ICA theory

In this appendix we give a short review of basic ICA theory. For a comprehensive treatment we refer to e.g. Ref. [4].

A.1. The linear ICA model

We consider the statistical model in which the observed signal $\mathbf{x}$ is given as a linear mixture of some sources $\mathbf{s}$

$$\mathbf{x} = \mathbf{A}\mathbf{s}. \tag{A.1}$$

The goal is to estimate the de-mixing $\mathbf{W}$, such that the vector $\mathbf{x}$ is transformed into

$$\mathbf{y} = \mathbf{W}\mathbf{x}, \tag{A.2}$$

yielding as mutually statistically independent components of $\mathbf{y}$ as possible (note that we have suppressed the time dependency). The estimation of $\mathbf{W}$ is done by presenting

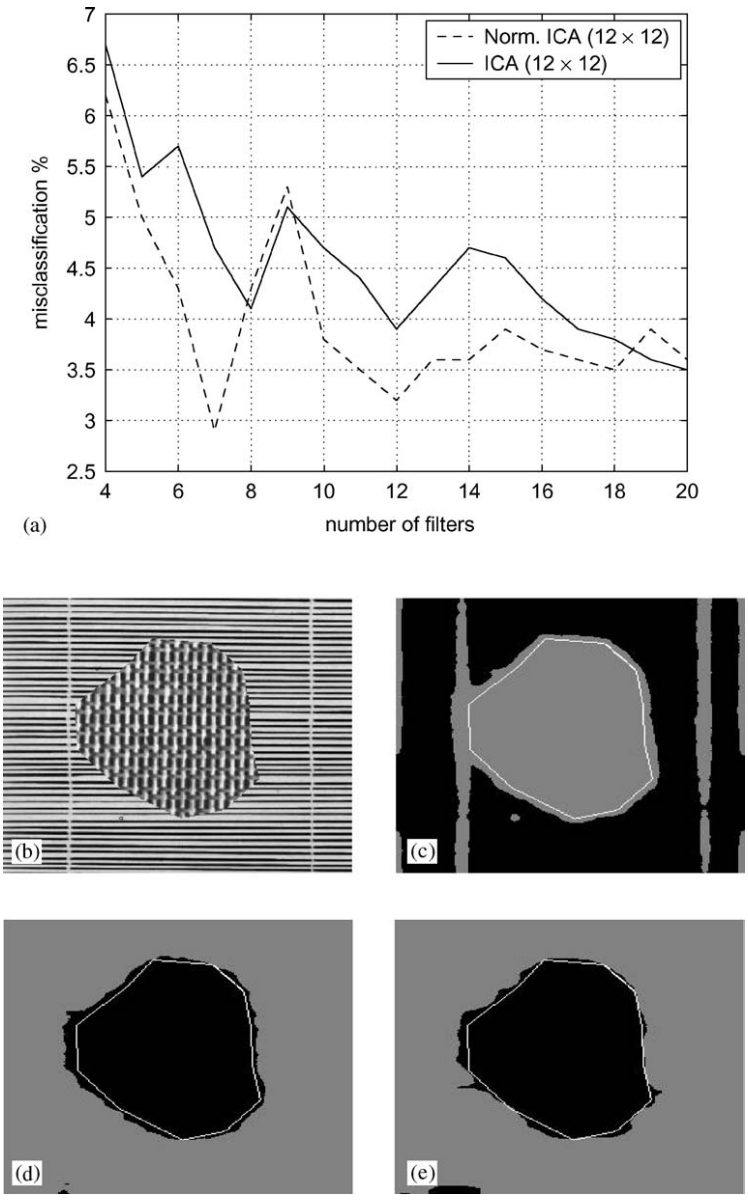


Fig. 9. Segmentation results. Textured image is composed of Brodatz textures $D53$ (inner pattern) and $D49$.

training vectors to an unsupervised algorithm optimizing independence of the components in $\mathbf{y}$. The vector $\mathbf{y}$ is thus an estimate of the true sources $\mathbf{s}$. The training vectors $\mathbf{x}$ are represented as samples of the random vector $\mathbf{X}$.

A.2. Preprocessing

Two preprocessing steps are common in ICA [21,25]. First, the data is centered,

$$\mathbf{x} := \mathbf{x} - E\{\mathbf{x}\}. \tag{A.3}$$

Thus, in ICA, we deal with zero mean data. Note that this means that $E\{\mathbf{s}\} = 0$.

The next step is to remove the second order statistical dependence in the data by whitening. Whitened data are uncorrelated and have unit variance. This is achieved by using a whitening matrix $\mathbf{V}$ such that

$$\mathbf{z} = \mathbf{V}\mathbf{x}, \quad \text{and} \quad E\{\mathbf{z}\mathbf{z}^{\mathrm{T}}\} = \mathbf{I}. \tag{A.4}$$

In our case we apply the common principal component analysis (PCA) whitening, i.e. $\mathbf{V}_{\mathrm{PCA}} = \mathbf{D}^{-1/2}\mathbf{E}^{\mathrm{T}}$, using the eigenvalue decomposition $\mathbf{E}\mathbf{D}\mathbf{E}^{\mathrm{T}} = E\{\mathbf{x}\mathbf{x}^{\mathrm{T}}\}$ of the covariance

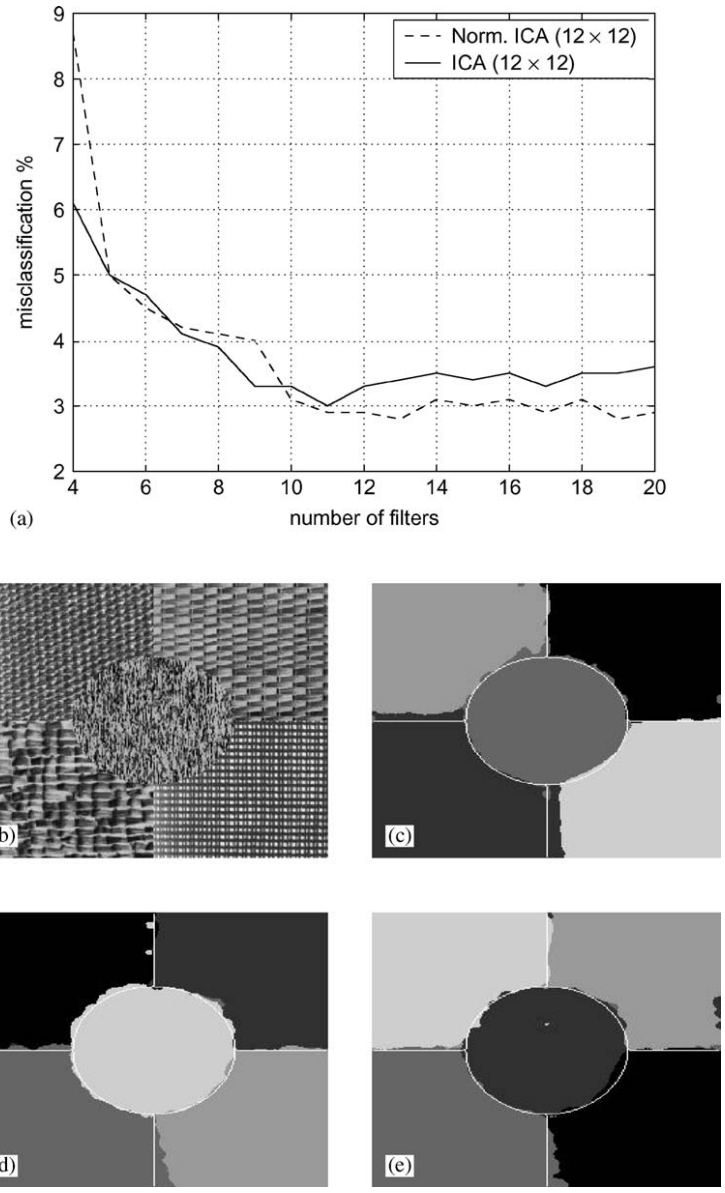


Fig. 10. Segmentation results. Textured image is composed of Brodatz textures $D77$, $D55$, $D84$, $D21$ (top left to bottom right) and $D24$ (inner pattern).

matrix of $\mathbf{x}$. $\mathbf{E}$ is here the matrix containing the eigenvectors of $E\{\mathbf{x}\mathbf{x}^{\mathrm{T}}\}$, and $\mathbf{D}$ holds the corresponding eigenvalues on its diagonal.

Having centered and whitened the data $\mathbf{x}$, the task is to find matrix $\mathbf{B}$, such that

$$\mathbf{y} = \mathbf{B}\mathbf{z} = \mathbf{B}\mathbf{D}^{-1/2}\mathbf{E}^{\mathrm{T}}\mathbf{x}. \tag{A.5}$$

Note that $\mathbf{W} = \mathbf{B}\mathbf{D}^{-1/2}\mathbf{E}^{\mathrm{T}}$. If the training data $\mathbf{x}$ is of dimension $(M \times 1)$, then in this basic ICA model, matrices $\mathbf{B}$, $\mathbf{D}$ and $\mathbf{E}$ are all of dimension $(M \times M)$, and the dimension of $\mathbf{y}$ is also $(M \times 1)$.

A.3. Independence optimization

There exist several algorithms performing ICA [58–63]. We have used Hyvärinens FastICA algorithm [62,63]. The FastICA algorithm constrains $\mathbf{B}$ to be orthogonal (orthonormal). It outputs the independent components one at a time by estimating matrix $\mathbf{B}$ row by row. That is, it first finds component $y_1 = \mathbf{b}_1^{\mathrm{T}}\mathbf{z}$, then $y_2 = \mathbf{b}_2^{\mathrm{T}}\mathbf{z}$ and so on. Here $\mathbf{b}_i^{\mathrm{T}}, i = 1, \ldots, M$, is the ith row of $\mathbf{B}$. Each new $\mathbf{b}_i^{\mathrm{T}}$ to be estimated is constrained to be orthogonal to all the previously estimated $\mathbf{b}_j^{\mathrm{T}}$, $j < i \leqslant M$.

Each vector $\mathbf{b}_i^{\mathrm{T}}$ is optimized in an iterative manner. Consider estimating the ith independent component, s_i. Denote the estimate of s_i at a certain iteration step by y_i. Thus $y_i = \hat{\mathbf{b}}_i^{\mathrm{T}}\mathbf{z}$, and by Eqs. (A.4) and (A.1),

$$y_i = \hat{\mathbf{l}}_i^{\mathrm{T}}\mathbf{s} = \sum_{j=1}^{M} \hat{l}_{ij} s_j, \tag{A.6}$$

where $\hat{\mathbf{l}}_i^{\mathrm{T}} = \hat{\mathbf{b}}_i^{\mathrm{T}}\mathbf{V}_{\mathrm{PCA}}\mathbf{A}$. Since the individual s_j, $j = 1,\ldots,M$, have unit variances by Eqs. (A.4) and (A.5), then by the central limit theorem, it is clear that y_i will be more Gaussian than any of the individual s_j, since a sum of independent variables with finite variance will produce a more Gaussian variable. The strategy employed is to force $y_i = \hat{\mathbf{b}}_i^{\mathrm{T}}\mathbf{z}$ towards maximum non-Gaussianity because this would in fact imply adjusting $\hat{\mathbf{l}}_i^{\mathrm{T}}$ until it has only one non-zero element (± 1), namely the element multiplied to the most non-Gaussian s_j. This enables us to estimate the ith independent component.

This discussion also brings to light why the true s_j, $j = 1,\ldots,M$ (except possibly one) need to have non-Gaussian statistics. If this is not the case, it would be impossible to separate the signals, since a sum of Gaussians remains Gaussian.

The iterative procedure is accomplished by updating each row $\mathbf{b}_i^{\mathrm{T}}$, $i = 1,\ldots,M$, of $\mathbf{B}$ by [62,63]

$$\mathbf{b}_i^{+} = E\{\mathbf{z}g_1(\mathbf{b}_i^{\mathrm{T}}\mathbf{z})\} - E\{g_1'(\mathbf{b}_i^{\mathrm{T}}\mathbf{z})\}\mathbf{b}_i, \tag{A.7}$$

followed by a normalization. In this equation $\mathbf{b}_i^{+}$ denotes the updated vector, and $g_1(u)$ is a non-linearity. We use $g_1(u) = \tanh(u)$ [62,63].

A.4. Reducing dimension

PCA enables us to optimally (in a squared error sense) reduce the dimension of the data, by using only the eigenvectors of the largest eigenvalues in the matrix $\mathbf{V}_{\mathrm{PCA}}$.

When reducing dimension from $\mathbf{x} \sim (M \times 1)$ to $\mathbf{y} \sim (N \times 1)$ where $N < M$, the model becomes

$$\mathbf{y}_N = \mathbf{B}\mathbf{z} = \mathbf{B}\mathbf{D}_N^{-1/2}\mathbf{E}_N^{\mathrm{T}}\mathbf{x}, \tag{A.8}$$

where $\mathbf{D}_N$ is the matrix containing the N largest eigenvalues, and $\mathbf{E}_N$ holds the corresponding eigenvectors. Note that in this case $\mathbf{B} \sim (N \times N)$, $\mathbf{D}_N \sim (N \times N)$ and $\mathbf{E}_N \sim (M \times N)$. Thus $\mathbf{y} \sim (N \times 1)$.

The reconstruction of image $\mathbf{x}$ using dimensionality reduction is $\hat{\mathbf{x}}_{\mathrm{PCA}} = \mathbf{E}_N\mathbf{z}_N$, with $\mathbf{z}_N = \mathbf{E}_N^{\mathrm{T}}\mathbf{x}$, and the mean square error between the image $\mathbf{x}$ and it's PCA reconstruction given by

$$E\{\|\mathbf{x} - \hat{\mathbf{x}}_{\mathrm{PCA}}\|^2|\}\} = E\{\|\mathbf{x} - \mathbf{E}_N\mathbf{E}_N^{\mathrm{T}}\mathbf{x}\|^2\} = \sum_{i=N+1}^{M} \lambda_i. \tag{A.9}$$

λ_i, $i = 1,\ldots,M$, are the eigenvalues sorted in decreasing order, and M is the original dimension of the data.

It is easy to show that the mean square error when using dimensionality reduction in the ICA transform is the same as in Eq. (A.9). The ICA transform (with reduced dimension) is given by Eq. (A.8). The reconstructed image, $\hat{\mathbf{x}}_{\mathrm{ICA}}$, is given as

$$\begin{aligned}\hat{\mathbf{x}}_{\mathrm{ICA}} &= \mathbf{A}\mathbf{y}_N = \mathbf{E}_N\mathbf{D}_N^{1/2}\mathbf{B}^{\mathrm{T}}\mathbf{y}_N \\ &= \mathbf{E}_N\mathbf{D}_N^{1/2}\mathbf{B}^{\mathrm{T}}\mathbf{B}\mathbf{D}_N^{-1/2}\mathbf{E}_N^{\mathrm{T}}\mathbf{x} = \mathbf{E}_N\mathbf{E}_N^{\mathrm{T}}\mathbf{x}.\end{aligned} \tag{A.10}$$

We utilize these results in Section 3.2 when calculating the normalized square error (NSE) between an image $\mathbf{x}$ and its estimate $\hat{\mathbf{x}}_{\mathrm{ICA}}$. We normalize the squared error with the sum of all eigenvalues of the covariance matrix $E\{\mathbf{x}\mathbf{x}^{\mathrm{T}}\}$ of the training data, such that the NSE is given by

$$NSE = \frac{\|\mathbf{x} - \hat{\mathbf{x}}_{\mathrm{ICA}}\|^2}{\sum_{i=1}^{M} \lambda_i} \in [0,1]. \tag{A.11}$$

References

[1] C. Jutten, J. Hérault, Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture, Signal Process. 24 (1991) 1–10.

[2] P. Comon, Independent component analysis—a new concept? Signal Process. 36 (1994) 287–314.

[3] C. Jutten, Source separation: from dusk till dawn, in: Proceedings of the Second International Workshop on Independent Component Analysis and Blind Source Separation (ICA2000), Helsinki, Finland, 2000, pp. 15–26.

[4] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, in: S. Haykin (Ed.), Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control, Wiley, New York, 2001.

[5] P. Comon, C. Jutten, J. Hérault, Blind separation of sources, part II: problems statement, Signal Process. 24 (1996) 11–20.

[6] E. Sorouchyari, Blind separation of sources, part III: stability analysis, Signal Process. 24 (1991) 21–29.

[7] T.P. Jung, C. Humphries, T.-W. Lee, S. Makeig, M.-J. McKeown, V. Iragui, T. Sejnowski, Extended ICA removes artifacts from electroencephalographic recordings, in: M.I. Jordan, M.J. Kearns, S.A. Solla (Eds.), Advances in Neural Information Processing Systems, Vol. 10, MIT Press, Cambridge, MA, 1998, pp. 894–900.

[8] R. Vigário, Extraction of ocular artifacts from EEG using independent component analysis, Electroencephalogr. Clin. Neurophysiol. 103 (3) (1997) 395–404.

[9] R. Vigário, V. Jousmäki, M. Hämälainen, R. Hari, E. Oja, Independent component analysis for identification of artifacts in magnetoencephalographic recordings, in: M.I. Jordan, M.J. Kearns, S.A. Solla (Eds.), Advances in Neural Information Processing Systems, Vol. 10, MIT Press, Cambridge, MA, 1998, pp. 229–235.

[10] R. Vigário, J. Särelä, V. Jousmäki, M. Hämäläinen, E. Oja, Independent component approach to the analysis of EEG and MEG recordings, IEEE Trans. Biomed. Eng. 47 (5) (2000) 589–593.

[11] R. Cristescu, J. Joutsensalo, J. Karhunen, E. Oja, A complexity minimization approach for estimating fading channels in CDMA communications, in: P. Pajunen, J. Karhunen (Eds.), Proceedings of the International Workshop

on Independent Component Analysis and Blind Signal Separation (ICA2000), Helsinki, Finland, June 2000, pp. 527–532.

[12] R. Cristescu, T. Ristaniemi, J. Joutsensalo, J. Karhunen, Blind separation of convolved mixtures for CDMA systems, in: M. Gabbouj, P. Kuosmanen (Eds.), Proceedings of the Tenth European Signal Processing Conference (EUSIPCO2000), Tampere, Finland, 2000, pp. 619–622.

[13] K. Pawelzik, K.-R. Müller, J. Kohlmorgen, Prediction of mixtures, in: C. von der Malsburg, W. von Seelen, J.C. Vorbrüggen, B. Sendhoff (Eds.), Proceedings of the International Conference on Artificial Neutral Networks (ICANN'96), Springer, Berlin, 1996, pp. 127–132.

[14] T. Eltoft, Ø. Kristensen, ICA and nonlinear times series prediction for recovering missing data segments in multivariate signals, in: T.-W. Lee, T.P. Jung, S. Makeig, T. Sejnowski (Eds.), Proceedings of the Third International Workshop on Independent Component Analysis and Blind Signal Separation (ICA2001), San Diego, USA, December 2001, pp. 716–721.

[15] S. Malaroiu, K. Kiviluoto, E. Oja, Time series prediction with independent component analysis, in: E. Tonkes, C. Tan, S. Sugden, K. Kumar (Eds.), Proceedings of the International Conference on Advanced Investment Technology, Gold Coast, Australia, 2000.

[16] K. Torkkola, Blind separation for audio signals—are we there yet? in: J.-F. Cardoso, C. Jutten, P. Lonbaton (Eds.), Proceedings of the First International Workshop on Independent Component Analysis and Source Separation (ICA'99), Aussois, France, 1999, pp. 239–244.

[17] F.M. Ham, N.A. Faour, J.C. Wheeler, Infrasound signal separation using independent component analysis, in Proceedings of the 21st Seismic Research Symposium: Technologies for Monitoring the Comprehensive Nuclear-Test-Ban Treaty, Las Vegas, NV, USA, 1999, Vol. 2, pp. 133–140.

[18] J.-F. Cardoso, C. Jutten, P. Loubaton, (Eds.), Proceedings of the First International Workshop on Independent Component Analysis and Signal Separation, Aussois, France, January 1999.

[19] P. Pajunen, J. Karhunen, (Eds.), Proceedings of the Second International Workshop on Independent Component Analysis and Blind Signal Separation, Helenski, Finland, June 2000.

[20] T.-W. Lee, T.P. Jung, S. Makeig, T. Sejnowski, (Eds.), Proceedings of the Third International Conference on Independent Component Analysis and Blind Signal Separation, San Diego, CA, USA, December 2001.

[21] A.J. Bell, T.J. Sejnowski, The "independent components" of natural scenes are edge filters, Vision Res. 37 (1997) 3327–3338.

[22] J. Hurri, Independent component analysis of image data, Master's Thesis, Helsinki University of Technology, 1997.

[23] J. Hurri, A. Hyvärinen, E. Oja, Wavelets and natural image statistics, in: M. Frydrych, J. Parkinnen, A. Visa (Eds.), Proceedings of the Scandinavian Conference on Image Analysis, Lappeenranta, Finland, 1997.

[24] J.H. van Hateran, A. van der Schaaf, Independent component filters of natural images compared with simple cells in primary visual cortex, Proc. R. Soc. Ser. B 256 (1998) 359–366.

[25] P. Hoyer, A. Hyvärinen, Independent component analysis applied to feature extraction from colour and stereo images, Network: Comput. Neural Systems 11 (3) (2000) 191–210.

[26] A. Hyvärinen, Sparse code shrinkage: denoising of nongaussian data by maximum likelihood estimation, Neural Comput. 11 (7) (1999) 1739–1768.

[27] R. Jenssen, T.A. Øigård, T. Eltoft, A. Hanssen, Sparse code shrinkage based on the normal inverse Gaussian density model, in: T.-W. Lee, T.P. Jung, S. Makeig, T. Sejnowski (Eds.), Proceedings of the Third International Workshop on Independent Component Analysis and Blind Signal Separation (ICA2001), San Diego, USA, December 2001, pp. 212–217.

[28] R.M. Haralick, K. Shanmugam, I. Distein, Textural features for image classification, IEEE Trans. Syst. Man Cybern. 3 (6) (1973) 610–621.

[29] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Academic Press, New York, 1999.

[30] L. Davis, S. Johns, J.K. Aggrawal, Texture analysis using generalized co-occurrence matrices, IEEE Trans. Pattern Anal. Mach. Intell. 1 (3) (1979) 251–259.

[31] H. Tamura, S. Mori, T. Yamawaki, Textural features corresponding to visual perception, IEEE Trans. Syst. Man Cybern. 8 (6) (1978) 460–473.

[32] J. Zhang, T. Tan, Brief review of invariant texture analysis methods, Pattern Recognition 35 (2002) 735–747.

[33] F. Cohen, Classification of rotated and scaled texture images using gaussian markov random field models, IEEE Trans. Pattern Anal. Mach. Intell. 13 (2) (1992) 198–202.

[34] K. Sivakumar, Morphologically constrained GRFs: applications to texture synthesis and analysis, IEEE Trans. Pattern Anal. Mach. Intell. 21 (2) (1999) 148–153.

[35] L. Fang, R.W. Picard, Periodicity, directionality, and randomness: Wold features for image modeling and retrieval, IEEE Trans. on Pattern Anal. and Mach. Intell. 18 (7) (1996) 722–733.

[36] L. Fang, R.W. Picard, Periodicity, directionality, and randomness: Wold features for perceptional pattern recognition, in: Proceedings of the International Conference on Pattern Recognition, Vol. 11, Jerusalem, October 1994, pp. 184–189.

[37] M. Tuceryan, A.K. Jain, in: C.H. Chen, L.F. Pau, P.S.P. Wang (Eds.), The Handbook of Pattern Recognition and Computer Vision, 2nd Edition, World Scientific, Singapore, 1998, pp. 207–248 (Chapter 21).

[38] F.W. Campbell, J.G. Robson, Application of Fourier analysis to the visibility of gratings, J. of Physiol. 197 (1968) 551–556.

[39] R.L. De Valois, D.G. Albrecht, L.G. Thorell, Spatial-frequency selectivity of cells in macaque visual cortex, Vision Res. 22 (1982) 545–559.

[40] T. Randen, J.H. Husøy, Filtering for texture classification: a comparative study, IEEE Trans. Pattern Anal. Mach. Intell. 21 (4) (1999) 291–310.

[41] K.I. Laws, Rapid texture identification, in: SPIE Image Processing for Missile Guidance, Vol. 238, SPIE, Bellingham, 1980, pp. 376–380.

[42] K.I. Laws, Textured Image Segmentation, Ph.D. Thesis, University of Southern California, 1980.

[43] J.M. Coggins, A.K. Jain, A spatial filtering approach to texture analysis, Pattern Recognition Lett. 3 (1985) 195–203.

[44] D.A. Pollen, S.F. Ronnen, Visual cortical neurons as localized spatial filters, IEEE Trans. Syst. Man Cybern. 13 (1983) 907–916.

[45] S. Marcelja, Mathematical description of the responses of simple cortical cells, J. Opt. Soc. Am. 70 (1980) 1297–1300.

[46] J.G. Daugman, Two dimensional spectral analysis of cortical receptive field properties, Vision Res. 20 (1980) 847–856.
[47] M.R. Turner, Texture discrimination by Gabor functions, Biol. Cybern. 55 (1986) 71–82.
[48] A.C. Bovik, M. Clark, W.S. Geisler, Multichannel texture analysis using localized spatial filters, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1990) 55–73.
[49] A.K. Jain, F. Farrokhnia, Unsupervised texture segmentation using Gabor filters, Pattern Recognition 24 (12) (1991) 1167–1186.
[50] O. Pichler, A. Teuner, B. Hosticka, A comparison of texture feature extraction using adaptive Gabor filtering, pyramidal and tree structured wavelet transforms, Pattern Recognition 29 (5) (1996) 733–742.
[51] A. Teuner, O. Pichler, B.J. Hosticka, Unsupervised texture segmentation of images using tunes matched Gabor filters, IEEE Trans. Image Process. 4 (6) (1995) 863–870.
[52] A. Laine, J. Fan, Texture classification by wavelet packet signatures, IEEE Trans. Pattern Anal. Mach. Intell. 15 (11) (1993) 1186–1191.
[53] T. Chang, C.-C.J. Kuo, Texture analysis and classification with tree-structured wavelet transform, IEEE Trans. Image Process. 2 (4) (1993) 429–441.
[54] N. Saito, R.R. Coifman, Local discriminant bases and their applications, J. Math. Imaging Vision 5 (4) (1995) 337–358.
[55] A. Laine, J. Fan, Frame representations for texture segmentation, IEEE Trans. Image Process. 5 (5) (1996) 771–780.
[56] M. Unser, Texture classification and segmentation using wavelet frames, IEEE Trans. Image Process. 4 (11) (1995) 1549–1560.
[57] T. Randen, J.H. Husøy, Multichannel filtering for image texture segmentation, Optical Eng. 33 (1994) 2617–2625.
[58] S. Amari, A. Cichocki, H. Yang, A new learning algorithm for blind source separation. in: Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 1996, pp. 757–763.
[59] A.J. Bell, T.J. Sejnowski, An information-maximization approach to blind separation and blind deconcolution, Neural Comput. 7 (1995) 1129–1159.
[60] J.-F. Cardoso, B.H. Laheld, Equivariant adaptive source separation, IEEE Tran. Signal Process. 44 (12) (1996) 3017–3030.
[61] A. Cichocki, R. Unbehauen, Robust neural networks with on-line learning for blind identification and blind separation of sources, IEEE Trans. Circuits Syst. 43 (11) (1996) 894–906.
[62] A. Hyvärinen, E. Oja, A fast fixed-point algorithm for independent component analysis, Neural Comput. 9 (1997) 1483–1492.
[63] A. Hyvärinen, Fast and robust fixed-point algorithms for independent component analysis, IEEE Trans. Neural Networks 10 (3) (1999) 626–634.
[64] H.B. Barlow, Unsupervised learning, Neural Comput. 1 (1989) 295–311.
[65] J.J. Atick, A.N. Redlich, What does the retina know about natural scenes? Neural Comput. 4 (1992) 308–320.
[66] H.B. Barlow, D.J. Tolhurst, Why do you have edge detectors? Opt. Soc. Am.: Tech. Dig. 23 (1992) 172.
[67] B.A. Olshausen, D.J. Field, Natural image statistics and efficient coding, Network: Comput. Neural Syst. 7(2) (1996).
[68] P. Brodatz, Texture: a photographic album for artists and designers, Dover, New York, 1996.
[69] T.P. Weldon, W.E. Higgins, Designing multiple Gabor filters for multitexture image segmentation, Opt. Eng. 38 (9) (1999) 1478–1489.
[70] M. Unser, M. Eden, Nonlinear operators for improving texture segmentation based on features extracted by spatial filtering, IEEE Trans. Syst. Man Cybern. 20 (1990) 804–815.
[71] T. Kasparis, D. Charalampidis, M. Georgiopoulos, J. Rolland, Segmentation of textured images based on fractals and image filtering, Pattern Recognition 34 (2001) 1963–1973.
[72] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[73] G.W. Milligan, M.C, Cooper, An examination of procedures for determining the number of clusters in a data set. Phychometrica 50 (1985) 159–179.

About the Author—ROBERT JENSSEN received the M. Eng. degree from the University of Tromsø, Norway in 2000. He specialized on denoising of images using the signal processing technique known as independent component analysis. Jenssen is currently pursuing a Ph.D. at the Group of Electrical Engineering at the Department of Physics, University of Tromsø. His current research interests are image and signal processing and pattern recognition.

About the Author—TORBJØRN ELTOFT received the Cand. Real. (M.S.) and Dr. Scient. (Ph.D.) degrees from the University of Tromsø, Norway, in 1981 and 1984, respectively. He specialized on the application of modern signal processing techniques in experimental ionospheric physics.

Since 1984 he has been working with remote sensing, with special emphasis on the non-linear SAR-imaging of ocean waves, and the scattering of microwaves from the ocean surface. He joined University of Tromsø, Department of Physics in 1988, where he is currently Professor at the Group of Electrical Engineering. His current research interests are remote sensing, image and signal processing, and artificial neural networks.

He received the year 2000 Outstanding Paper Award in Neural Networks by the IEEE Neural Networks Council.

Chapter 7

Future Research Directions

In this thesis, we have shown that Parzen window-based estimators for several quadratic information measures have dual interpretations as Mercer kernel-based measures and graph theoretic measures. Furthermore, there is a close link to spectral methods. This means for example that the clustering algorithm we presented in paper 1, which was based on standard optimization techniques like Lagrange optimization, is closely related to (graph) spectral clustering. This was not realized at the time the algorithm was derived. We mention that in [Jenssen et al., 2004a] a brief comparison of some of the Cauchy-Schwarz-based clustering algorithms presented in this thesis was performed.

These close links between different machine learning schemes should be more closely examined in future work. In particular, it should be investigated whether tools known from different schemes could be combined into more powerful machine learning algorithms. For example, we have seen that Parzen window size selection procedures can be helpful in making spectral clustering algorithms automatic with respect to the kernel size which determines affinites between data points. In future work we will study whether our spectral clustering algorithm (paper 2) may be implemented without actually carrying out the data mapping. We believe this to be possible, since it is a Mercer kernel-based method. Other Mercer kernel-based methods such as the support vector machine *implicitly* operate in the Mercer kernel feature space using the "kernel-trick." The link to graph theory has not been explored in depth in this thesis. Graph theory is an old field of research, and may potentially add useful insights to machine learning.

In an excellent survey on statistical learning theory Jain et al. [2000] wrote: "The topic of probabilistic distance measures is currently not as important as 20 years ago, since it is very difficult to estimate density functions in high dimensional feature spaces. Instead, the complexity of classification procedures and the resulting accuracy have gained a large interest. The curse of dimensionality as well as the danger of overtraining are some of the consequences of a complex classifier. It is now understood that these problems can, to some extent, be circumvented using regularization, or can even be completely resolved by a proper design of classification procedures. The study of support vector machines has largely contributed to this understanding."

Although this statement is true in many respects, it may be that the probabilistic distance measures play a more important role after all, and that a key component to see this is Parzen windowing. The topic of information theoretic regularization, which we touched upon in paper 3, could be helpful in understanding some of the issues raised in this quote.

We also note that the ICA filter bank segmentation procedure is not necessarily constrained to textured images alone. It should be applicable to any kind of image data. This is a topic for future research. In this respect, we want to mention one example which has already appeared in the literature in the paper [Lopez-Martinez et al., 2004], in which our ICA segmentation procedure was used in landmine detection.

Further, in conclusion of this section, we mention two recent conference papers which points to research topics which we think needs further attention. These papers are

1) R. Jenssen, D. Erdogmus, J. C. Principe and T. Eltoft, "The Laplacian Spectral Classifier," in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Philadelphia, USA, March 18 - 23, 2005.

2) J.-W. Xu, D. Erdogmus, R. Jenssen and J. C. Principe, "An Information Theoretic Perspective to Kernel Independent Component Analysis," in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Philadelphia, USA, March 18 - 23, 2005.

These papers are included. A special thanks goes to Jian-Wu Xu for allowing the second paper to appear in this thesis.

THE LAPLACIAN SPECTRAL CLASSIFIER *

Robert Jenssen†*, Deniz Erdogmus*‡*, Jose C. Principe*‡ *and Torbjørn Eltoft*†

† University of Tromsø, N-9037 Tromsø, Norway
‡ University of Florida, Gainesville, 32611 FL., USA

ABSTRACT

We develop a novel classifier in a kernel feature space defined by the eigenspectrum of the Laplacian data matrix. The classification cost function is derived from a distance measure between probability densities. The Laplacian data matrix is obtained based on a training set, while test data is mapped to the kernel space using the Nyström routine. In that space, the test data is classified based on the angle between the test point and the training data class means. We illustrate the performance of the new classifier on synthetic and real data.

1. INTRODUCTION

Spectral methods for multivariate data analysis are emerging as powerful tools, mostly based on their practical successes, for example in clustering [1]. Spectral methods are typically based on a kernel matrix of pairwise relationships between the samples, from which a more useful data representation can be derived by utilizing its eigenvalue decomposition, or eigenspectrum. Until recently, only those points used to calculate the kernel matrix have been possible to represent in the kernel feature space. Therefore, spectral classifiers have been slow to emerge since these have to be able to represent successively new data points in the kernel feature space. Recently, it was shown how the map new data points into the feature space by using the Nyström routine [2].

In this paper, we propose a new spectral classifier based on the Laplacian pdf distance, which is introduced as a clustering cost function in a recent paper by the current authors [3]. The Laplacian pdf distance exhibits a connection to Mercer kernel based learning theory via the Parzen window technique for density estimation. In a kernel feature space defined by the eigenspectrum of the Laplacian data matrix, this distance measures the cosine of the angle between the class mean vectors. Interestingly, in [3] it was shown that when the prior probabilities of the classes are roughly equal, minimizing the Laplacian pdf distance corresponds to minimizing the probability of error. However, if the prior probabilities are unequal, the Laplacian pdf distance will act as a risk function, emphasizing to classify correctly the least probable class.

Quite importantly, based on the Parzen method, an optimal spectral data transformation can be obtained. We propose to learn the optimal Laplacian data matrix based on a training data set. Hence, the transformation to the kernel feature space is defined by the eigenspectrum of that matrix. In the kernel space, we compute the means of the transformed training data. A test data set, which is to be classified, is mapped to the kernel space by means of the Nyström routine. Based on the Laplacian pdf distance in the kernel space, a spectral classifier is developed. The angle between a test point and the class means is computed. Thereafter, the test point is assigned to the class yielding the smallest such angle.

For the convenience of the reader, we briefly review the theory behind the Laplacian pdf distance in section 2. The material presented here is a compressed version of [3]. We only consider the two-class case, even though multiclass generalizations can easily be made. In section 3, we develop the novel Laplacian spectral classifier. Thereafter, in section 4, we present some experimental studies of the proposed method. Finally, in section 5, we make our concluding remarks.

2. THE LAPLACIAN PDF DISTANCE

2.1. Mercer kernel-based feature spaces

In Mercer kernel-based learning algorithms a nonlinear mapping is potentially performed as

$$\mathbf{\Phi} : R^d \rightarrow \mathcal{F}$$
$$\mathbf{x} \rightarrow \mathbf{\Phi}(\mathbf{x}) = [\sqrt{\lambda_1}\phi_1(\mathbf{x}), \sqrt{\lambda_2}\phi_2(\mathbf{x}), \dots]^T, \quad (1)$$

where the λ_i's and the ϕ_i's are the eigenvalues and eigenfunctions of a Mercer kernel. Hence, the data $\mathbf{x}_1, \dots, \mathbf{x}_N \in R^d$ is mapped into $\mathbf{\Phi}(\mathbf{x}_1), \dots, \mathbf{\Phi}(\mathbf{x}_N) \in \mathcal{F}$. The Mercer kernel computes an inner product in the feature space, that is, $k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{y}) \rangle$ [4]. In practice, the mapping (1) is approximated based on the eigenspectrum

*THIS WORK WAS PARTIALLY SUPPORTED BY NSF GRANT ECS-0300340.

of the $(N \times N)$ kernel matrix, $\mathbf{K}$, with elements $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j),\ i,j = 1,\ldots,N$, as

$$\Phi(\mathbf{x}_i) \approx [\sqrt{\tilde{\lambda}_1}e_{1i}, \ldots, \sqrt{\tilde{\lambda}_N}e_{Ni}]^T. \quad (2)$$

where $\tilde{\lambda_j}$ is the jth eigenvalue and e_{ji} denotes the ith element of the jth eigenvector of the matrix $\mathbf{K}$.

In [2] it was shown that an estimate of the eigenfunction at a new point, $\mathbf{y}$, can be obtained by he following interpolatory formula, denoted the Nyström routine

$$\phi_j(\mathbf{y}) \approx \frac{\sqrt{N}}{\tilde{\lambda}_j} \sum_{i=1}^{N} e_{ji} k(\mathbf{y}, \mathbf{x}_i). \quad (3)$$

2.2. The Laplacian PDF distance as a kernel feature space cost function

Assume that a data set consists of two clusters. Associate the probability density function $p(\mathbf{x})$ with one of the clusters, and the density $q(\mathbf{x})$ with the other cluster. Let $f(\mathbf{x})$ be the overall probability density function of the data set. A distance measure between the two pdfs can be expressed as

$$D_L = -\log \frac{\langle p, q \rangle_f}{\sqrt{\langle p, p \rangle_f \langle q, q \rangle_f}} \geq 0. \quad (4)$$

where the f^{-1} weighted inner product between $p(\mathbf{x})$ and $q(\mathbf{x})$ is defined as $\langle p, q \rangle_f \equiv \int p(\mathbf{x}) q(\mathbf{x}) f^{-1}(\mathbf{x}) d\mathbf{x}$. By defining the two functions $h(\mathbf{x}) = f^{-\frac{1}{2}}(\mathbf{x}) p(\mathbf{x})$ and $g(\mathbf{x}) = f^{-\frac{1}{2}}(\mathbf{x}) q(\mathbf{x})$, the argument of the $\log$ in (4) can be expressed as

$$L = \frac{\int h(\mathbf{x}) g(\mathbf{x}) d\mathbf{x}}{\sqrt{\int h^2(\mathbf{x}) d\mathbf{x} \int g^2(\mathbf{x}) d\mathbf{x}}}. \quad (5)$$

The distance between the two pdfs is greater the smaller (5) is. Assume that we have available the iid training data points $\{\mathbf{x}_i\},\ i = 1,\ldots,N_1$, drawn from $p(\mathbf{x})$, which is the density of class C_1, and the iid $\{\mathbf{x}_j\},\ j = 1,\ldots,N_2$, drawn from $q(\mathbf{x})$, the density of C_2. The union of these two classes constitutes the overall data set. The relevant functions can be estimated based on the Parzen window density estimation technique as

$$\hat{h}(\mathbf{x}) = \frac{1}{N_1} \sum_{i=1}^{N_1} f^{-\frac{1}{2}}(\mathbf{x}_i) W_{\sigma_1^2}(\mathbf{x}, \mathbf{x}_i),$$

$$\hat{g}(\mathbf{x}) = \frac{1}{N_2} \sum_{j=1}^{N_2} f^{-\frac{1}{2}}(\mathbf{x}_j) W_{\sigma_2^2}(\mathbf{x}, \mathbf{x}_j),$$

and $\hat{f}(\mathbf{x}) = \frac{1}{N} \sum_{k=1}^{N} W_{\sigma^2}(\mathbf{x}, \mathbf{x}_k)$, where W is a Gaussian kernel function whose width is determined by the σ^2-parameter in each case. By inserting these estimates into (5), it was shown that it has an equivalent expression in a Mercer kernel feature space as

$$L = \frac{\langle \mathbf{m}_{1_f}, \mathbf{m}_{2_f} \rangle}{||\mathbf{m}_{1_f}|| ||\mathbf{m}_{2_f}||},$$

where $\mathbf{m}_{i_f} = \frac{1}{N_i} \sum_{l=1}^{N_i} \mathbf{\Phi}_f(\mathbf{x}_l),\ i = 1, 2$, that is, the sample mean of the ith class in feature space. The Gaussian Parzen kernel and the Mercer kernel is in fact equivalent in this case. This cost function is quite interesting. It measures the distance between the two classes in the feature space. In that space, the distance is solely based on the means of the classes. The distance is given by the cosine of the *angle* between the class mean vectors.

The mapping $\mathbf{\Phi}_f$ was shown to be determined by the eigenspectrum of the matrix $\mathbf{K}_f$. This matrix can be written as $\mathbf{K}_f = \mathbf{D}^{-\frac{1}{2}} \mathbf{K} \mathbf{D}^{-\frac{1}{2}}$. Here, $\mathbf{K}$ is the kernel matrix with elements $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = W_{(\sigma_t^2 + \sigma_s^2)}(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathbf{x}_i \in C_t, \mathbf{x}_j \in C_s$, for $t, s \in \{1, 2\}$. Furthermore, $\mathbf{D} = \text{diag}(d_1, \ldots, d_N)$, where $d_i = \hat{f}(\mathbf{x}_i) = \frac{1}{N} \sum_{j=1}^{N} W_{\sigma^2}(\mathbf{x}_i, \mathbf{x}_j)$. In fact, $\mathbf{K}_f$ is the *Laplacian data matrix*.

A key point of this paper is that σ_1, σ_2 and σ, can be determined automatically from the training set by optimal Parzen kernel size selection. Thus, the matrix $\mathbf{K}_f$ can also be determined automatically, and so can the mapping to the kernel feature space.

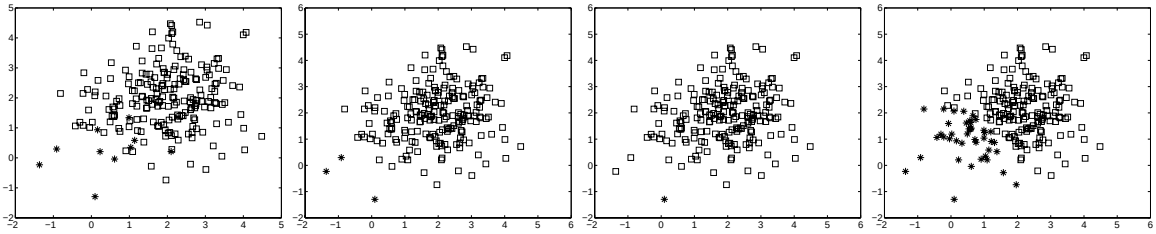
Many approaches have been proposed in order to optimally determine the size of the Parzen window, given a finite sample data set. Silverman [5] discussed this problem, using the mean integrated square error (MISE) between the estimated an the actual pdf as the optimality metric, and proposed the following formula

$$\sigma_{\text{opt}} = \sigma_X \left\{ 4N^{-1}(2d+1)^{-1} \right\}^{\frac{1}{d+4}}, \quad (6)$$

where d is the dimensionality of the data and $\sigma_X^2 = d^{-1} \sum_i \mathbf{\Sigma}_{X_{ii}}$, where $\mathbf{\Sigma}_{X_{ii}}$ are the diagonal elements of the sample covariance matrix.

3. A NOVEL SPECTRAL CLASSIFIER

In this section, we discuss a novel method for developing a spectral classifier based on the Laplacian pdf distance. We have available a labeled training data set. For each of the classes, the optimal Parzen kernel size is determined by (6). The optimal kernel size for the overall data set is also determined by the same formula. Now, the optimal data transformation into the kernel feature space can be performed by (2), after having constructed $\mathbf{K}_f$. Note that the dimensionality of the data in the kernel space equals the number of training data patterns. In that space, the training class mean vectors can be calculated, which can be used to determine



(a) Test data - labeled.

(b) Bayes classification of test data.

(c) Kernel density classification of test data.

(d) Spectral classification of test data.

Fig. 1. Result of classifying a data set consisting of two Gaussian classes with very different prior probabilities.

the distance between the classes. This is the training phase of the classifier. For a test data set, which is to be classified, one data point, $\mathbf{y}$, at a time is mapped into the feature space by (3). We use a Gaussian kernel also in (3), where the kernel size, σ, is based on the overall training data set, since we don't know which class $\mathbf{y}$ belongs to. Thereafter, the angle between $\Phi(\mathbf{y})$ and each of the training class mean vectors is computed. Finally, $\mathbf{y}$ is classified to the class for which that angle is the smallest. In summary, the proposed classifier has the following steps

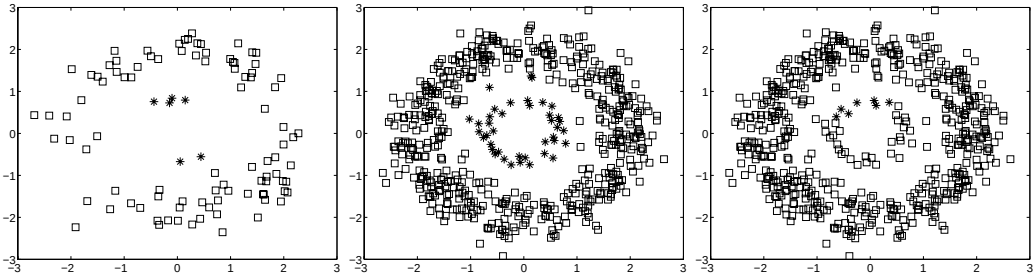
1. Determine σ_1, σ_2 and σ using (6) in each case.
2. Calculate $\mathbf{K}$, $\mathbf{D}$ and $\mathbf{K}_f = \mathbf{D}^{-\frac{1}{2}}\mathbf{K}\mathbf{D}^{-\frac{1}{2}}$.
3. Eigendecompose $\mathbf{K}_f$, and compute $\Phi(\mathbf{x}_i) \approx [\sqrt{\tilde{\lambda}_1}e_{1i}, \ldots, \sqrt{\tilde{\lambda}_N}e_{Ni}]^T, \forall i$.
4. Find $\mathbf{m}_1$ and $\mathbf{m}_2$.
5. for i = 1 : number of test points
   - Map $\mathbf{y}_i$ the the kernel space by (3).
   - Find the angle θ_1 between $\Phi(\mathbf{y}_i)$ and $\mathbf{m}_1$ and the angle θ_2 between $\Phi(\mathbf{y}_i)$ and $\mathbf{m}_2$.
   - Classify: $\mathbf{y}_i \in C_1$ if $\theta_1 < \theta_2$, else $\mathbf{y}_i \in C_2$.

4. EXPERIMENTAL RESULTS

Experiment 1. In the first classification experiment, we classify data points originating from two Gaussian distributions. The purpose is to illustrate the risk function property of the Laplacian spectral classifier. Both distributions have the same spherical covariance structure with unit variance. The mean vector of class one in the input space is $\mu_1 = [2\ 2]^T$. The mean vector of the second class is $\mu_2 = [0.6\ 0.6]^T$. The training data is constructed such that class one is represented by 100 data points, compared to only 5 data points from class two. Hence, $P_1 \approx 0.95$, while $P_2 \approx 0.05$. This means that the two clusters have overlap and that their prior probabilities are very different. Based on this training data set, the new spectral classifier is trained. For comparison, we construct a traditional Gaussian Bayes classifier. Since the covariances of the Gaussian classes are equal, this classifier produces a linear boundary between the classes. We also train a traditional Parzen kernel Bayes classifier [6], using (6) to determine the appropriate kernel size for each of the two classes. Recall that the Bayes classifier is in theory optimal with respect to the probability of error. The test data set is drawn from the same Gaussian distributions as for the training set. The data set consists of 200 data points from class one, and 10 from class two.

A scatter plot of the labeled test data set is shown in Fig. 1 (a). The squares indicate class one, and the stars class two. It can be seen that the data sets overlap, such that classification errors are unavoidable. The classification result using the Gaussian Bayes classifier is shown in Fig. 1 (b). It performs very well in terms of classification errors. It misclassifies only 7 data points. All the misclassified data points belong to class two. The Parzen kernel Bayes classifier performs worse, only detecting one of the class two data points, as shown in Fig. 1 (c). The spectral classifier obtains the result shown in Fig. 1 (d). The result is significantly different from that obtained by the Bayes classifiers. It classifies correctly 9 of the class two data points. However, it also erroneously assigns 31 class one data points to class two. One class two data point is wrongly assigned to class one. Clearly, the Laplacian spectral classifier emphasizes more to classify correctly the least probable class, i.e. the class two data points in this case. This property may be useful in many applications.

The results presented in this experiment vary somewhat depending on the training data and the test data, which is drawn at random from the Gaussian distributions. However, these differences are small, and the result presented here is representative for most cases. It should be mentioned that for $P_1 \approx P_2$, the classifiers perform almost equally good.



(a) Training data.

(b) Spectral classifi cation of test data.

(c) Kernel density classifi cation of test data.

Fig. 2. Result of classifying a data set consisting of two ring-shaped classes.

Experiment 2. The purpose of the second experiment is to show that the spectral classifier can handle highly irregular data shapes. Fig. 2 (a) shows the labeled training data set, which consists of two ring-shaped classes. There are 100 training samples, 6 from the inner-most ring and 94 from the outer-most ring. Fig. 2 (b) shows the spectral classification result for a test set consisting of 844 test samples, drawn from the same ring-shaped distributions. The classification result is nearly completely correct, for this very challenging data set. The classification result is fairly stable over repeated experiments. For comparison, Fig. 2 (c) shows the classification result using the Parzen kernel Bayes classifier. Again, it has problems with the sparse class.

Experiment 3. In this experiment, we classify a breast-cancer data set into the two classes *benign* and *malignant*. The purpose is to show that the proposed classifier also performs well on a real data set of higher dimensionality than for the previous two data sets. The Wisconsin Breast-Cancer (WBC) database is the source of this dataset, which consists of 683 data points (444 benign and 239 malignant). WBC is a nine-dimensional dataset. For the training data, 100 data points were selected at random from the data set. We performed the classification 20 times, each time selecting different training data at random. The test set consisted in each case of 583 data patterns. The average correct classification rate was 96.0%, with a standard deviation of 0.01%. The Parzen kernel Bayes classifier performs almost equally good in this case, probably because the prior probabilities are not extremely different.

5. CONCLUSIONS

We have presented a new fully automatic (no user-specified parameters) spectral classifier based on the Laplacian pdf distance. The training data set is optimally mapped to the feature space using the eigenspectrum of the Laplacian data matrix. New data points are mapped to the feature space by the Nyström routine, where they are classified based on the angle with the means of the transformed training data. The new classifier has been shown to perform well on irregular and real data. Also, it exhibits the interesting property that it emphasizes to classify correctly the least probable data points.

As for most kernel-based methods, proper kernel size selection may be problematic in very high-dimensional data spaces.

6. REFERENCES

[1] Y. Weiss, "Segmentation Using Eigenvectors: A Unifying View," in *International Conference on Computer Vision*, 1999, pp. 975–982.

[2] C. Williams and M. Seeger, "Using the Nyström Method to Speed Up Kernel Machines," in *Advances in Neural Information Processing Systems 13*, MIT Press, 2001, pp. 682–688.

[3] R. Jenssen, D. Erdogmus, J. C. Principe, and T. Eltoft, "The Laplacian PDF Distance: A Cost Function for Clustering in a Kernel Feature Space," in *Advances in Neural Information Processing Systems 17*, MIT Press, 2005.

[4] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[5] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London, 1986.

[6] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 1999.

AN INFORMATION-THEORETIC PERSPECTIVE TO KERNEL INDEPENDENT COMPONENTS ANALYSIS

Jian-Wu Xu[1], Deniz Erdogmus[2], Robert Jenssen[3], Jose C. Principe[1]

[1]CNEL, Dept. of Electrical and Computer Engineering, University of Florida, USA
[2]Dept. of Computer Science and Engineering, Oregon Graduate Institute, OHSU, USA
[3]Dept. of Physics, University of Tromsø, Norway

ABSTRACT

In this paper, we investigate the intriguing relationship between information-theoretic learning (ITL), based on weighted Parzen window density estimator, and kernel-based learning algorithms. We prove the equivalence between kernel independent component analysis (KERNEL ICA) and Cauchy-Schwartz (C-S) independence measure. This link gives a theoretical motivation for the selection of the Mercer kernel, based on density estimation. Demonstrating this equivalence requires introducing a weighted kernel density estimator, a modification of Parzen windowing. We also discuss the role of the weights in the weighted Parzen windowing and KERNEL ICA.

1. INTRODUCTION

Kernel-based learning algorithms have been developed in the machine learning community during the last decades. With the introduction of support vector machine (SVM) theory [1], kernel Fisher discriminant (KFD) [2], and kernel principal component analysis (KPCA) [3], one is able to obtain nonlinear algorithms from linear ones in a simple and elegant way. Kernel-based algorithms are nonlinear versions of linear algorithms where the data has been nonlinearly transformed to a high dimensional feature space where we only need to compute the inner product via the kernel function. The attractiveness of kernel-based algorithms resides in their elegant treatment of nonlinear problems and efficiency for high-dimensional problems. Kernel methods have been successfully applied to time series prediction, DNA and protein analysis, optical pattern and object recognition [4].

Recently, Bach *et al* proposed a class of kernel-based algorithms for independent component analysis (ICA) that utilize contrast functions based on canonical correlations in a reproducing kernel Hilbert space, named KERNEL ICA [5]. The KERNEL ICA is based on novel kernel-based measures of dependence and can be computed efficiently. Minimizing these criteria results in flexible and robust ICA algorithms. One problem with all kernel methods is that it is not theoretically clear how to choose the best kernel function. The most commonly used kernel function is the Gaussian kernel, but it is still an open question how to select the width of Gaussian kernel in general.

In parallel to the developments in kernel-based methods research, independently a research topic called *information-theoretic learning* (ITL) has emerged [6], where kernel-based density estimators form the essence of this learning paradigm. Information-theoretic learning is a signal processing technique that combines information theory and adaptive systems. ITL utilizes information theory as a criterion to update the structure of adaptive system in order to achieve a certain performance. By utilizing Renyi's measure of entropy and approximations to the Kullback-Leibler probability density divergence, ITL is able to extract information beyond second-order statistics directly from data in a non-parametric manner. Information-theoretic learning has achieved excellent results on a number of learning scenarios, e.g. blind source separation [7, 8], time series prediction [9].

In this paper, we examine the KERNEL ICA from an information-theoretic learning perspective. We show that KERNEL ICA is equivalent to minimizing the Cauchy-Schwartz independence measure, when estimated via weighted Parzen windowing, though they have different normalizations. Based on the discussions in this paper, we conjecture that the kernel-based algorithms, including the KERNEL ICA, which are expressed in terms of inner products in the kernel feature space, are in fact learning implicitly by using non-parametric estimates of probability densities in the input space. This new view gives a geometrical interpretation for KERNEL ICA and theoretical criterion for choosing the Mercer kernel used in the kernel-based algorithms such that it would lead to a relatively accurate estimate when used as the Parzen windowing in density estimation. Before we proceed to that, we first show that how the most widely used ITL cost functions, when estimated by Parzen windowing, can

This work was supported by NSF grant ECS-0300340.

be expressed in terms of inner products in a reproducing kernel Hilbert space.

This paper is organized as follows. We review the basic theory of nonlinear kernel feature space and the KERNEL ICA in section 2. Cauchy-Schwartz independence measure is introduced in section 3. Afterwards, in section 4, we show how some of ITL cost functions can be written into quantities defined in the Hilbert feature space via Parzen windowing and prove the equivalence between the KERNEL ICA and Cauchy-Schwartz independence measure. Furthermore, we discuss the role of weights, used in weighted Parzen windowing for probability density estimation and the corresponding kernel space.

2. KERNEL ICA

Kernel-based learning algorithms use the following idea: via a nonlinear mapping

$$\Phi : \Re^t \to \mathcal{F} \qquad \mathbf{x} \to \Phi(\mathbf{x}) \tag{1}$$

the data in the input space $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N} \in \Re^t$ is mapped to a potentially much higher dimensional feature space $\mathcal{F}$. Instead of considering the given learning problem in input space $\Re^t$, one can deal with $\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \ldots, \Phi(\mathbf{x}_N)$ in feature space $\mathcal{F}$. When the learning algorithms can be expressed in terms of inner products, this nonlinear mapping becomes particular interesting and useful since one can employ the *kernel trick* to compute the inner products in the feature space via *kernel functions* without knowing the exact nonlinear mapping Φ. This way of addressing the given learning problems allows one to obtain nonlinear algorithms from linear ones in a simple and elegant manner. In essence, by Mercer's theorem [10], the eigen-decomposition of a positive function (the kernel) is utilized to define the following inner product for the transformation space:

$$\kappa_\sigma(x - x') = \sum_{k=1}^{\infty} \lambda_k \varphi_k(x) \varphi_k(x') = \langle \Phi(x), \Phi(x') \rangle \tag{2}$$

where $\langle \cdot, \cdot \rangle$ denotes an inner product, the φ_k's are the eigen-functions of the kernel and λ_k's are the associated eigenvalues. The KERNEL ICA presented by Bach *et al* is a new method to ICA based on a kernel-measure of independence [5]. KERNEL ICA assumes a reproducing kernel Hilbert space (RKHS) $\mathcal{F}$ with kernel $\kappa(x, x')$ and feature map $\Phi(x) = \kappa(\cdot, x)$. Then the $\mathcal{F}$-correlation is defined as the maximal correlation between the two random variables $f_1(x_1)$ and $f_2(x_2)$, where f_1 and f_2 range over $\mathcal{F}$:

$$\begin{aligned} \rho &= \max_{f_1, f_2} corr(f_1(x_1), f_2(x_2)) \\ &= \max_{f_1, f_2} \frac{\mathrm{cov}(f_1(x_1), f_2(x_2))}{\sqrt{(\mathrm{var}\, f_1(x_1))(\mathrm{var}\, f_2(x_2))}} \end{aligned} \tag{3}$$

Clearly, if the random variables x_1 and x_2 are independent, then the $\mathcal{F}$-correlation is zero. Moreover, the converse is also true provided that the set $\mathcal{F}$ is large enough. This means that $\rho = 0$ implies x_1 and x_2 are independent.

In order to obtain a computationally tractable implementation of $\mathcal{F}$-correlation, the *reproducing property* of RKHS is used to estimate the $\mathcal{F}$-correlation,

$$f(x) = \langle \Phi(x), f \rangle = \langle \kappa(\cdot, x), f \rangle \tag{4}$$

Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be the linear spaces spanned by the Φ-images of the data samples, then f_1 and f_2 can be decomposed into two parts, i.e.

$$f_1 = \sum_{k=1}^{N} \alpha_1^k \Phi(x_1^k) + f_1^{\perp}, \quad f_2 = \sum_{k=1}^{N} \alpha_2^k \Phi(x_2^k) + f_2^{\perp} \tag{5}$$

where $f_1^{\perp}$ and $f_2^{\perp}$ are orthogonal to $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively. Using the empirical data to approximate the population value, the $\mathcal{F}$-correlation can be estimated as

$$\hat{\rho} = \max_{\alpha_1, \alpha_2 \in \Re^N} \frac{\alpha_1^T K_1 K_2 \alpha_2}{\sqrt{(\alpha_1^T K_1^2 \alpha_1)(\alpha_2^T K_2^2 \alpha_2)}} \tag{6}$$

where K_1 and K_2 are the Gram matrices associated with the data sets $\{x_1^k\}$ and $\{x_2^k\}$ defined as $(K_i)_{a,b} = \kappa(x_i^a, x_i^b)$.

In the paper [5], Bach *et al* used a regularized version for the expression (6) by penalizing the RKHS norms of f_1 and f_2 in the denominator because (6) is not a consistent estimator in general. The regularized estimator has the same independence characterization property of the $\mathcal{F}$-correlation as (6), since it is the numerator, $\alpha_1^T K_1 K_2 \alpha_2$, in the $\mathcal{F}$-correlation that characterizes the independence property of two random variables. The difference between the direct estimator (6) and the regularized version is only the normalization. This also can be seen in section 4 when we prove the equivalence between the KERNEL ICA and Cauchy-Schwartz (C-S) independence measure.

3. CAUCHY-SCHWARTZ INDEPENDENCE MEASURE

In this section, we introduce the Cauchy-Schwartz (C-S) independence measure, which has been utilized as a cost function in independent component analysis (ICA) [7] and clustering [11].

In information theory, mutual information is a quantity

that characterizes the divergence between two random variables. A well-known divergence measure is the Kullback-Leibler distance

$$K(f,g)=\int f(x)\log\frac{f(x)}{g(x)}dx \qquad (7)$$

where *f(x)* and *g(x)* are two probability density functions (pdf). The Kullback-Leibler measure is difficult to evaluate in practice, without imposing simplifying assumptions about the data, since numerical methods are required to evaluate the integrals. In order to elegantly integrate the non-parametric pdf estimation via Parzen windowing [12], Principe *et al* proposed a new pdf distance measure based on Cauchy-Schwartz inequality between two vectors [6]. Thus we can evaluate pdf distance measure without making any parametric assumptions about the underlying pdfs.

Based on the Cauchy-Schwartz inequality, $\|\mathbf{x}\|^2\|\mathbf{y}\|^2 \geq (\mathbf{x}^T\mathbf{y})^2$, we can write $-\log \mathbf{x}^T\mathbf{y}\Big/\sqrt{\|\mathbf{x}\|^2\|\mathbf{y}\|^2} \geq 0$. Replacing the inner product between vectors by inner product between pdfs, we can define the Cauchy-Schwartz independence measure as

$$D_{CS}(f,g)=-\log\frac{\int f(x)g(x)dx}{\sqrt{(\int f^2(x)dx)(\int g^2(x)dx)}} \qquad (8)$$

Notice that $D_{CS}(f,g)\geq 0$ and the equality holds if and only if *f(x)=g(x)*. For two random variables X_1 and X_2, with marginal pdfs $f_1(x_1)$ and $f_2(x_2)$ and joint pdf $f_{1,2}(x_1,x_2)$, the Cauchy-Schwartz independence measure becomes

$$\begin{aligned} D_{CS}(f_1,f_2) &= -\log J \\ &= -\log\frac{\iint f_{1,2}(x_1,x_2)f_1(x_1)f_2(x_2)dx_1dx_2}{\sqrt{(\iint f_{1,2}^2(x_1,x_2)dx_1dx_2)(\iint f_1^2(x_1)f_2^2(x_2)dx_1dx_2)}} \\ &= -\log\frac{E[f_1(x_1)f_2(x_2)]}{\sqrt{E[f_{1,2}(x_1,x_2)]E[f_1(x_1)]E[f_2(x_2)]}} \end{aligned} \qquad (9)$$

As can be seen from above that $D_{CS}(f_1,f_2)\geq 0$. If and only if the two random variables are statistically independent, then $D_{CS}(f_1,f_2)=0$. Hence minimization of Cauchy-Schwartz independence measure leads to minimization of mutual information between two random variables. This is exactly the idea that Cauchy-Schwartz independence measure can be used as a criterion to characterize independence for ICA in [7].

In the next section, we will proceed to prove that the KERNEL ICA is equivalent to Cauchy-Schwartz independence measure, when estimated via weighted Parzen windowing.

4. EQUIVALENCE BETWEEN KERNEL ICA AND C-S INDEPENDENCE MEASURE

In this section, we first show how some widely used cost functions in information-theoretic learning can be estimated directly from data sample through Parzen windowing method. More importantly, these cost functions can be written in terms of inner products in a reproducing kernel Hilbert space, where the Mercer kernel is the windowing function used in Parzen density estimation. Then the proof of equivalence between KERNEL ICA and C-S independence measure will follow naturally.

4.1. ITL Cost Functions in the Kernel Space

One of the most commonly used cost functions in information-theoretic learning is the quadratic Renyi's entropy because it can be easily integrated with the Parzen window estimator [6], thus provides a simple way to estimate the entropy directly from the data samples.

Given the pdf *f(x)* for a random variable *X*, quadratic Renyi's entropy is defined as

$$H(X)=-\log\int f^2(x)dx=-\log E[f(x)] \qquad (10)$$

Since logarithm is a monotonic function, the quantity of interest is its argument $V(X)=\int f^2(x)dx$, which is called *information potential*. For a given pdf *f(x)*, a non-parametric asymptotically unbiased and consistent estimator is given by [12]

$$\hat{f}(x)=\frac{1}{N}\sum_{i=1}^{N}\kappa(x,x_i) \qquad (11)$$

where $\kappa(.)$ is called the Parzen window, or kernel. It is often chosen to be the Gaussian kernel though other kernels are also available, e.g., polynomial kernels. Then approximating the expectation by sample mean, we can estimate the *information potential* direct from data

$$V(X)=\frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}\kappa(x_i-x_j) \qquad (12)$$

Notice that $\kappa(.)$ is a Gaussian kernel function, Hence we can employ (2) to rewrite (14) as

$$\begin{aligned} V(X) &= \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}\left\langle\Phi(x_i),\Phi(x_j)\right\rangle \\ &= \left\langle\frac{1}{N}\sum_{i=1}^{N}\Phi(x_i),\frac{1}{N}\sum_{j=1}^{N}\Phi(x_j)\right\rangle=\left\|\mathbf{m}^{\Phi}\right\|^2 \end{aligned} \qquad (13)$$

where $\mathbf{m}^{\Phi}$ is the mean vector of the transformed data. Thus, the quadratic information potential turns out to be the inner product of the mean vector of the nonlinearly transformed data in the Hilbert kernel space.

4.2. Equivalence of KERNEL ICA and C-S Independence Measure

To prove the equivalence between KERNEL ICA and Cauchy-Schwartz independence measure, we use weighted Parzen windowing. For a given marginal pdf *f(x)*, the weighted Parzen windowing density estimator is given by

$$\hat{f}(x) = \frac{1}{A}\sum_{i=1}^{N} \alpha_i \kappa(x, x_i) \qquad (14)$$

When Cauchy-Schwartz independence measure (10) is used as a contrast function in ICA, it should be minimized so that the mutual information between random variables is also minimized. As logarithm is a monotonic function, minimizing the C-S quantity is equivalent to maximizing its argument. Approximating the expectation by sample mean in (10) and estimating pdfs with weighed Parzen windowing, we can get

$$\hat{J} = \max_{\alpha_1,\alpha_2 \in \Re^N} \frac{\alpha_1^T K_1 K_2 \alpha_2}{\sqrt{(V)(1^T K_1 \alpha_1)(1^T K_2 \alpha_2)}} \qquad (15)$$

where $V = \sum_{j=1}^{N}\sum_{i=1}^{N} \alpha_1^i \kappa(x_1^i, x_1^j)\kappa(x_2^i, x_2^j)\alpha_2^i$,

$1=[1,1,\ldots.]^T$, and $(K_i)_{a,b} = \kappa(x_i^a, x_i^b)$.

Comparing expressions (15) and (6), we notice that they have same numerators and different normalizations. As we already pointed out in section 2, the numerator in KERNEL ICA characterizes the independence measure of two random variables whereas the denominator gives the certain normalization. Hence we conclude that the Cauchy -Schwartz independence measure, estimated via weighed Parzen windowing, is equivalent to the KERNEL ICA.

4.3. Role of the Weights

Recently we showed that the SVM is related to ITL and non-parametric pdf estimation via weighted Parzen windowing. The weights in the Parzen windowing there is associated with the support vectors in SVM [13]. In the Cauchy-Schwartz independence measure with weighted Parzen windowing estimation, we notice that those weights are associated with the coordinates of nonlinear function f_1 and f_2 in the linear spaces $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively.

6. CONCLUSIONS

In this paper, we discuss the connection between information-theoretic learning (ITL), based on the weighted Parzen window density estimator that we have introduced. We demonstrated that the KERNEL ICA algorithm evaluates the independence between the separated outputs through a measure that is equivalent to the C-S mutual information estimated using the weighted Parzen windowing procedure. This discussion reveals an intriguing duality between the Mercer kernels and Parzen windowing (i.e., kernel density estimation). This duality provides a theoretical criterion for selecting the Mercer kernel in kernel-methods for machine learning and signal processing.

7. REFERENCES

[1] V. N. Vapnik, *The Nature of Statistical Learning Theory,* Springer-Verlag, NY, 1995.

[2] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller, " Fisher Discriminant Analysis with Kernels," Proc. NNSP'99*,* pp. 41-48, Piscataway, NJ, 1999.

[3] B. Schölkopf, A. J. Smola, and K. R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation*, vol. 10, pp. 1299-1319, 1998.

[4] B. Schölkopf and A. J. Smola, *Learning with Kernels.* MIT Press, 2001.

[5] F. R. Bach, M. I. Jordan, "Kernel Independent Component Analysis", *Journal of Machine Learning Research,* vol. 3, pp. 1-48, 2002.

[6] J.C. Principe, D. Xu, J. Fisher, "Information Theoretic Learning," in *Unsupervised Adaptive Filtering*, (Ed. S. Haykin), Wiley, NY, 2000.

[7] D. Xu, J. C. Principe, J. Fisher III, H. –C, Wu, "A novel measure for independent component analysis (ICA)," *Proc. ICASSP'98*, vol. 2, pp. 12-15, 1998.

[8] K. E. Hild, D. Erdogmus, J. C. Principe, "Blind Source Separation using Renyi's Mutual Informaiton," *IEEE Signal Processing Letters*, vol. 8, no. 6, pp. 174-176, 2001.

[9] D. Erdogmus, J.C. Principe, "Generalized Information Potential Criterion for Adaptive System Training," *IEEE Trans. Neural Networks*, vol. 13, no. 5, pp. 1035-1044, 2002.

[10] J. Mercer, " Functions of Positive and Negative Type and their Connection with the Theory of Integral Equations," *Philos. Trans. Roy. Soc. London,* vol. A. pp. 415-446, 1909.

[11] R. Jenssen, J. C. Principe and T. Eltoft, "Cauchy-Schwartz pdf Divergence Measure for non-Parametric Clustering," *Proc. NORSIG'03,* Bergen, Norway, 2003.

[12] E. Parzen, "On Estimation of a Probability Density Function and Mode", in *Time Series Analysis Papers*, Holden-Day, CA, 1967.

[13] R. Jenssen, D. Erdogmus, J. Principe, T. Eltoft, "Towards a Unification of Information Theoretic Learning and Kernel Methods," *Proc. MLSP'04*, Sao Luis, Brazil, 2004.

Appendix A

In this Appendix, we discuss a relationship between the quadratic information measures, estimated using Parzen windowing, and graph theory. The relationship between the Cauchy-Schwarz divergence and graph theory has already been presented in paper 2, but is repeated for completeness. It is assumed that the reader recalls the expressions for the Parzen window-based estimators for the quadratic information measures, which have been derived throughout papers 1-3.

A set of points in an arbitrary feature space can be represented as a weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. The nodes of the graph are the points in feature space, i.e. node k corresponds to point $\mathbf{x}_k,\ k = 1, \ldots, N$. An edge is formed between every pair of nodes. The weight on each edge, for example between nodes k and k' will be denoted $k(\mathbf{x}_k, \mathbf{x}_{k'})$, and is a function of the similarity between the two nodes. Oftentimes, an exponential function is used to define similarity, that is

$$k(\mathbf{x}_k, \mathbf{x}_{k'}) = \exp\left\{-\frac{||\mathbf{x}_k - \mathbf{x}_{k'}||^2}{2\sigma_{\mathcal{G}}^2}\right\}, \tag{7.1}$$

where $\sigma_{\mathcal{G}}$ is the width of the exponential function, associated with the graph $\mathcal{G}$.

A graph may also be partitioned into two subgraphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, by removing edges in $\mathcal{G}$. We denote the weights on these edges which have to be removed by $k(\mathbf{x}_i, \mathbf{x}_j)$, where node i is in $\mathcal{G}_1$ and node j is in $\mathcal{G}_2$, for $i = 1, \ldots, N_1$ and $j = 1, \ldots, N_2$.

Renyi's Quadratic Entropy and Graph Theory

The total sum of all the edges in a graph is called the volume of the graph, that is

$$Vol(\mathcal{G}) = \sum_{k,k'=1}^{N,N} k(\mathbf{x}_k, \mathbf{x}_{k'}). \tag{7.2}$$

Recall that the *information potential* can be written as

$$V(f) = \frac{1}{N^2} \sum_{k,k'=1}^{N,N} k(\mathbf{x}_k, \mathbf{x}_{k'}), \tag{7.3}$$

$k(\mathbf{x}_k, \mathbf{x}_{k'}) = W_{2\sigma^2}(\mathbf{x}_k, \mathbf{x}_{k'})$, and $W_{2\sigma^2}$ is the Gaussian Parzen window. Hence, it is clear that the Information Potential, and therefore Renyi's quadratic entropy, is connected to the volume of a graph, since $V(\mathbf{x}) = N^2 Vol(\mathcal{G})$.

Cauchy-Schwarz Divergence and Graph Theory

We will illustrate the connection between the Cauchy-Schwarz divergence and graph theory by introducing a concept from graph theory known as the *cut*.

The *cut* is a way to measure the cost of a partitioning of the graph $\mathcal{G}$ into the pieces $\mathcal{G}_1$ and $\mathcal{G}_2$, by summing the weights of the edges which have to be removed in $\mathcal{G}$, that is

$$Cut(\mathcal{G}_1, \mathcal{G}_2) = \sum_{i,j=1}^{N_1,N_2} k(\mathbf{x}_i, \mathbf{x}_j). \tag{7.4}$$

A small *cut* indicates that the graph $\mathcal{G}$ naturally splits into two dense subgraphs. The minimum *cut*-criterion has therefore been upheld as a natural criterion for data partitioning. However, the minimum *cut*-criterion is not unproblematic, since it can be seen that the criterion is minimized in the case that only one node constitute one of the subgraphs, and all the other nodes constitute the other subgraph.

Note that in our Parzen window framework the *cut* naturally arises, since

$$\int \hat{p}(\mathbf{x})\hat{q}(\mathbf{x})d\mathbf{x} = \frac{1}{N_1 N_2} \sum_{i,j=1}^{N_1,N_2} k(\mathbf{x}_i, \mathbf{x}_j). \tag{7.5}$$

Recall that the *information cut* (Cauchy-Schwarz divergence) can be written as

$$\begin{aligned} IC(p,q) &= \frac{\sum_{i,j=1}^{N_1,N_2} k(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{\sum_{i,i'=1}^{N_1,N_1} k(\mathbf{x}_i, \mathbf{x}_{i'}) \sum_{j,j'=1}^{N_2,N_2} k(\mathbf{x}_j, \mathbf{x}_{j'})}} \\ &= \frac{Cut(\mathcal{G}_1, \mathcal{G}_2)}{\sqrt{Vol(\mathcal{G}_1)Vol(\mathcal{G}_2)}}. \end{aligned} \tag{7.6}$$

The numerator of this expression is exactly the *cut*. The denominator acts as a normalizing quantity, since the *cut* is divided by the square root of the product of the volumes of each subgraph.

Integrated Squared Error Divergence and Graph Theory

The Parzen window-based estimator for the integrated squared error divergence is given by

$$\begin{aligned} \widehat{ISE}(p,q) &= \frac{1}{N_1^2} \sum_{i,i'=1}^{N_1,N_1} k(\mathbf{x}_i, \mathbf{x}_{i'}) + \frac{1}{N_2^2} \sum_{j,j'=1}^{N_2,N_2} k(\mathbf{x}_j, \mathbf{x}_{j'}) - 2\frac{1}{N_1 N_2} \sum_{i,j=1}^{N_1,N_2} k(\mathbf{x}_i, \mathbf{x}_j) \\ &= N_1^2 Vol(\mathcal{G}_1) + N_2^2 Vol(\mathcal{G}_2) - 2N_1 N_2 Cut(\mathcal{G}_1, \mathcal{G}_2). \end{aligned} \tag{7.7}$$

Hence, also the integrated squared error criterion is related to graph theory through the subgraph volumes and the *cut*-cost.

Appendix B

In this Appendix, we include two conference papers. The topic of both papers is clustering using the Cauchy-Schwarz divergence, and they are thus related to paper 1 and paper 2. They are included for completeness, to show that the CS divergence may be optimized using other approaches than those presented in paper 1 and paper 2. The work presented here has not been extended into comprehensive manuscripts, and are not regarded as important parts of this thesis, as compared to papers 1-4.

1) R. Jenssen, J. C. Principe and T. Eltoft, "Cauchy-Schwartz Divergence Measure for Non-Parametric Clustering," Proceedings of IEEE Norway Section Signal Processing Symposium (cd-rom), Bergen, Norway, October 2-4, 2003.

2) R. Jenssen T. Eltoft and J. C. Principe, "Information Theoretic Spectral Clustering," Proceedings of International Joint Conference on Neural Networks, Budapest, pages 111-116, Hungary, July 25-29, 2004.

CAUCHY-SCHWARTZ PDF DIVERGENCE MEASURE FOR NON-PARAMETRIC CLUSTERING

Robert Jenssen[1,2], *Jose C. Principe*[2] *and Torbjørn Eltoft*[1]

[1]Department of Physics
University of Tromsø, Norway

[2]Computational NeuroEngineering Laboratory
University of Florida, USA

ABSTRACT

We propose a new cost function for clustering based on the Cauchy-Schwartz (CS) inequality. This cost function is obtained by replacing inner products between vectors with inner products between probability density functions (pdf's), in the CS inequality expression. Combined with Parzen pdf estimation, this provides us with a non-parametric information-theoretic cluster evaluation function that we refer to as the CS divergence. We propose a novel method for maximization of the CS divergence for clustering, and present results on both artificial data and real data.

1. INTRODUCTION

In exploratory data analysis it is often desirable to perform an unsupervised classification of data patterns into different subsets, such that patterns within each subset are *alike* and patterns across subsets are *not alike*, according to some criterion. This problem is known as clustering [1]. Clustering has become an important tool in areas such as data mining [2], image segmentation [3], signal compression [4] and machine learning [5].

The two main approaches to clustering can be divided into the parametric and the non-parametric methods. In parametric methods some knowledge about the clusters' structure is assumed. The most famous and popular such method is McQueen's K-means algorithm [6], which implicitly assumes Gaussian cluster distributions. It minimizes a sum-of-squares cost function, equivalent to variance minimization, and thus fails if the cluster distributions are not hyper-elliptical. Often, however, there is no a-priori knowledge about the data structure. In such cases it is more natural to adopt non-parametric approaches, which make no model assumptions, such as e.g. single-link or complete-link hierarchical clustering [1]. Implicitly, usually these methods also rely on a minimum variance criterion as the clustering metric [7].

In order to capture data structure beyond second order statistics, information-theoretic clustering metrics, such as entropy, mutual information and Kullback-Leibler divergence [8], appear as an appealing alternative. Information theory has been used in clustering by Watanabe [9], and by several other researchers, e.g. [10, 11, 7]. The major problem of clustering based on information-theoretic measures has been the difficulty to evaluate the metric without imposing unrealistic parametric assumptions about the data distributions.

Recently, Principe et al. [12] proposed a pdf divergence measure that lends itself nicely to non-parametric estimation via Parzen windowing [13]. This pdf divergence measure is based on the Cauchy-Schwartz inequality between two vectors. Combined with a Gaussian kernel in Parzen pdf estimation, the Cauchy-Schwartz pdf divergence was utilized for blind source separation and pose estimation in synthetic aperture radar imagery [12]. Later, Gokcay and Principe [14] used a somewhat similar measure for clustering, with positive results.

In this paper we show that the Cauchy-Schwartz pdf divergence measure can be utilized as a cost function for non-parametric clustering. The remainder of the paper is organized as follows. In the next section we define the Cauchy-Schwartz pdf divergence measure, and show how it can be estimated directly from the available data in a non-parametric fashion. In section 3 we propose a novel method for maximization of this cost function for clustering. In section 4 we illustrate the performance of the proposed method using some artificial data and some real data. Finally, in section 5 we make our concluding remarks.

This work was partially supported by grants ECS-0300340 and EIA-0135946

R. Jenssen was supported by a scholarship from the University of Tromsø

Corresponding author is R. Jenssen. Phone: (+1) 352-392-2682, Fax: (+1) 352-392-0044, Email: robertj@phys.uit.no

2. CAUCHY-SCHWARTZ PDF DIVERGENCE MEASURE

Based on the Cauchy-Schwartz inequality; $||\mathbf{x}||^2\ ||\mathbf{y}||^2 \geq (\mathbf{x}^T\mathbf{y})^2$, the following holds;

$$-\log \frac{\mathbf{x}^T\mathbf{y}}{\sqrt{||\mathbf{x}||^2\ ||\mathbf{y}||^2}} \geq 0. \qquad (1)$$

In the spirit of Eq. (1), we define the following divergence measure between the two pdf's $p(\mathbf{x})$ and $q(\mathbf{x})$ [12];

$$D_{CS}(p,q) = -\log \frac{\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}}{\sqrt{\int p^2(\mathbf{x})d\mathbf{x}\ \int q^2(\mathbf{x})d\mathbf{x}}}. \qquad (2)$$

The logarithm is included to make the notation consistent with that of other divergence measures. This measure can be regarded as an approximation of the Kullback-Leibler divergence between the two pdf's. It is always positive, it vanishes if and only if $p(\mathbf{x}) = q(\mathbf{x})$ and it is symmetric. Maximizing the divergence between $p(\mathbf{x})$ and $q(\mathbf{x})$ is equivalent to minimizing the argument of the logarithm.

Assume that we estimate $p(\mathbf{x})$ based on the data points in cluster $C_1 = \{\mathbf{x}_i\}$, $i = 1,\ldots,N_p$, and $q(\mathbf{x})$ based on $C_2 = \{\mathbf{x}_j\}$, $j = 1,\ldots,N_q$. By the Parzen [13] method

$$\begin{aligned} \hat{p}(\mathbf{x}) &= \frac{1}{N_p}\sum_{i=1}^{N_p} G(\mathbf{x}-\mathbf{x}_i, \sigma^2\mathbf{I}), \\ \hat{q}(\mathbf{x}) &= \frac{1}{N_q}\sum_{j=1}^{N_q} G(\mathbf{x}-\mathbf{x}_j, \sigma^2\mathbf{I}), \end{aligned} \qquad (3)$$

where we have used a symmetric Gaussian kernel, $G(\mathbf{x}, \mathbf{\Sigma})$, with a covariance matrix given by $\mathbf{\Sigma} = \sigma^2\mathbf{I}$.

Now, we define the membership function M_{ij}, which equals one iff the data points $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to different clusters, and zero if not. Furthermore, we define the membership functions $M_{C_{1_{ij}}}$ and $M_{C_{2_{ij}}}$. If $\mathbf{x}_i$ and $\mathbf{x}_j$ both belong to C_1, then $M_{C_{1_{ij}}} = 1$, but zero if not. Likewise for $M_{C_{2_{ij}}}$ with respect to C_2. By substituting (3) into (2), and utilizing the properties of the Gaussian kernel [12], we can estimate $D_{CS}(p,q) = -\log V_{CS}(p,q)$ in a non-parametric fashion, where;

$$\begin{aligned} &\hat{V}_{CS}(p,q) = \qquad (4) \\ &= \frac{\frac{1}{N_pN_q}\sum_{i,j}^{N_p,N_q} G_{ij,2\sigma^2\mathbf{I}}}{\sqrt{\frac{1}{N_p^2}\sum_{i,i'}^{N_p,N_p} G_{ii',2\sigma^2\mathbf{I}}\ \frac{1}{N_q^2}\sum_{j,j'}^{N_q,N_q} G_{jj',2\sigma^2\mathbf{I}}}} \\ &= \frac{\frac{1}{2}\sum_{i,j}^{N,N} M_{ij}G_{ij,2\sigma^2\mathbf{I}}}{\sqrt{\sum_{i,j}^{N,N} M_{C_{1_{ij}}}G_{ij,2\sigma^2\mathbf{I}}\ \sum_{i,j}^{N,N} M_{C_{2_{ij}}}G_{ij,2\sigma^2\mathbf{I}}}}, \end{aligned}$$

where $\sum_{i,j}^{N,N} G_{ij,2\sigma^2\mathbf{I}} = \sum_{i=1}^{N}\sum_{j=1}^{N} G_{ij,2\sigma^2\mathbf{I}}$, $N = N_p + N_q$ and $G_{ij,2\sigma^2\mathbf{I}} = G(\mathbf{x}_i - \mathbf{x}_j, 2\sigma^2\mathbf{I})$.

In the case of multiple clusters, C_k, $k = 1,\ldots,K$, we extend the previous definition as follows;

$$\hat{V}_{CS}(p,q) = \frac{\frac{1}{2}\sum_{i,j}^{N,N} M_{ij}G_{ij,2\sigma^2\mathbf{I}}}{\sqrt{\prod_{k=1}^{K}\sum_{i,j}^{N,N} M_{C_{k_{ij}}}G_{ij,2\sigma^2\mathbf{I}}}}. \qquad (5)$$

At this point we note a particularly interesting feature of the Cauchy-Schwartz divergence. Based on Eq. (5), it is easily shown that $\hat{D}_{CS}(p,q)$ is in fact an estimate of Renyi's quadratic entropy [12] calculated between samples of different clusters, subtracted (half) the sum of Renyi's quadratic entropy of each individual cluster. This means that in order for $D_{CS}(p,q)$ to be large, the sum of the entropies of the individual clusters must be small, while at the same time the entropy across the clusters must be large. This makes perfect sense.

There is also another interesting interpretation of the Cauchy-Schwartz divergence. Actually, the Gaussian kernel used in the Parzen pdf estimation can be regarded a Mercer kernel [15], performing a nonlinear data transformation into some high dimensional feature space, which increases the probability of linear separability of the clusters in the transformed space. Inner products in the feature space are implicitly computed in the input space by use of the kernel. This means that $G_{ij,2\sigma^2\mathbf{I}} = \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$, where $\Phi(\cdot)$ denotes the mapping from input space to feature space. Minimizing $V_{CS}(p,q)$ is equivalent to minimizing the sum of inner products across clusters in feature space, while at the same time maximizing the product of the sums of inner products within clusters.

In conclusion, Eq. (5) provides us with an information-theoretic cluster evaluation function, capable of capturing data structure beyond mere second order statistics, as many traditional clustering algorithms are restricted to.

3. MAXIMIZING THE DIVERGENCE

In this section we propose a novel method for maximization of the Cauchy-Schwartz pdf divergence. The method we propose here has close resemblance to the approach taken by Jenssen et al. [10] in their entropy-based algorithm. The main idea is to "seed" a number of small initial clusters in the data set, grow the clusters until all patterns have been labeled, and then re-cluster the members of the "worst" cluster, thus reducing the number of clusters by one. This procedure is repeated until the predefined number of clusters is reached.

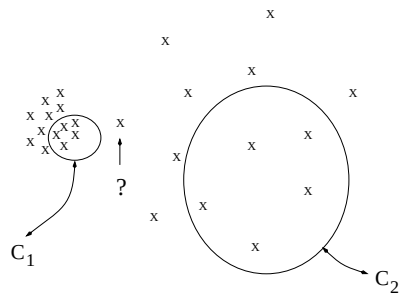


Figure 1: Assigning a data pattern to a cluster.

3.1. Growing Clusters

Consider the situation depicted in Fig. 1. A set of patterns, or feature vectors, is distributed in feature space. Initially a subset of the feature vectors have been assigned to cluster C_1 or C_2. These are shown as the encircled points. The problem of clustering is now to decide whether a new pattern $\mathbf{x}$ (pointed to by the question mark) should be assigned to C_1 or C_2. Our solution to this problem is very simple: Assign $\mathbf{x}$ to C_1 if $D_{CS}(C_1+\mathbf{x},C_2) > D_{CS}(C_1,C_2+\mathbf{x})$, and to C_2 if the opposite is true.

Hence, in the general case of having initial clusters $C_k,\ k=1,\ldots,K$, assign $\mathbf{x}$ to cluster C_i if

$$\max_i D_{CS}(C_1,\ldots,C_i+\mathbf{x},\ldots,C_K), \qquad (6)$$

for $i=1,\ldots,K$.

This approach to clustering is both intuitive and simple. However, at this point two questions arise: How to initially cluster a subset of the data? And, how to decide which pattern to be clustered next?

Initially we "seed" K_{init} clusters in the data set. This is done by randomly selecting K_{init} "seed" patterns, one at a time. The first "seed" pattern and it's $N_{\text{init}}-1$ nearest neighbors constitute the first cluster. Thereafter a new pattern among the unlabeled patterns is randomly selected, which together with it's $N_{\text{init}}-1$ closest unlabeled neighbors constitute the second cluster. This process is repeated until K_{init} clusters have been formed.

The next pattern to be clustered can be selected in several ways.

- Randomly among the unlabeled patterns - this has the disadvantage that early clustering of points far from the initial clusters can make the clustering process unstable.

- As the unlabeled pattern closest to a cluster prototype - this approach makes the clustering more stable. If we choose to have just one cluster prototype per cluster, it is typically the cluster mean. At the other extreme, all patterns in a cluster are regarded prototypes.

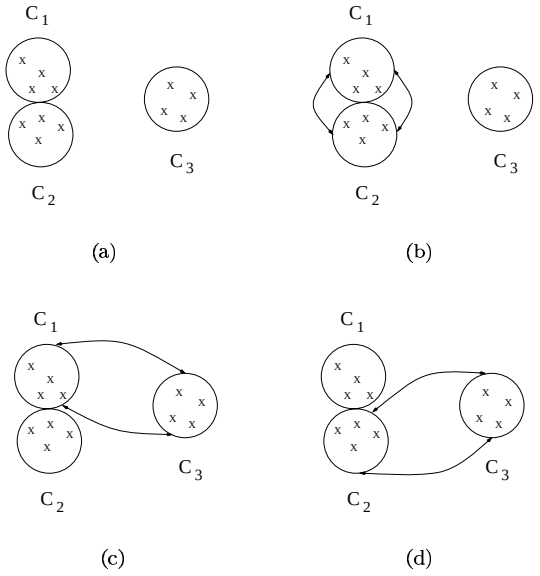


Figure 2: Identifying the "worst cluster" using the Cauchy-Schwartz pdf divergence. In this example C_1 or C_2 is the "worst cluster."

3.2. Cluster Evaluation

In Fig. 2 we have illustrated how the CS divergence can be used to identify the "worst cluster", whose members subsequently are re-clustered. In (a) a data set has been partitioned into three clusters. We proceed by eliminating one cluster at a time, and calculate the CS divergence based on the remaining clusters in each case. To be more specific, by eliminating, we mean that the members of the eliminated cluster are considered unlabeled, not contributing to the value of the CS divergence. For example, in (b) C_3 is eliminated, and the CS divergence based on C_1 and C_2 alone is calculated, as indicated by the arrows. Likewise in (c) and (d).

The "worst cluster" is now selected as the cluster that when eliminated, results in the largest CS divergence based on the remaining clusters, because this means that the remaining clusters are the most separated clusters. In the situation depicted in Fig. 2, this method results in either C_1 or C_2 being identified as the "worst cluster".

3.3. Complexity and instability

Usually the CS divergence clustering algorithm performs better by choosing the next pattern to be clustered as the unlabeled pattern closest to a cluster prototype. This clearly increases the complexity of the algorithm compared to random selection. In our implementation of the algorithm, the total complexity is always less than $O(N^2)$ at any iteration.

Because the initial clusters are "seeded" randomly, the clustering result can differ when clustering the same data set several times. The CS divergence calculated for the final clustering in each case can provide us with a good indication of which random initialization provided the best result.

4. PERFORMANCE STUDIES

In this section we test the performance of our novel algorithm on four data sets, two artificially created and two real. In all experiments the data have been normalized to have a range $[-1, 1]$. This is mainly done in order to have some control over the parameter σ. It doesn't affect the structure of the data.

The first data set is shown in Fig. 3. This data set consists of one spherical cluster and three elongated clusters, with a total of 550 patterns. To show that K-means, based on second order statistics, is incapable of producing a correct labeling of this data set, we ran it 10 times, and display in Fig. 3 (b) the best result. The clusters are represented by different symbols. Fig. 3 (a) shows the best result (but also typical result) obtained by the CS method for $\sigma = 0.06$. In this case we "seed" $K_{init} = 20$ clusters, each with $N_{init} = 10$ members. The next pattern to be clustered is the one closest to some labeled pattern. The result is very satisfying, yielding only two errors. The thick lines identifies the erroneously labeled patterns, and indicates that they actually belong to the cluster marked by squares. Our experiments show that these two patterns are always wrongly labeled by the CS method. However, this is to be expected, since by visual examination of the data set, a human would probably also assign these two patterns to the lower horizontal cluster. Next, we investigate the sensitivity of the method wrt. σ and K_{init}. First, we perfom an experiment where we keep K_{init} and N_{init} fixed at 20 and 10, respectively. We apply the CS method 10 times for a wide range of σ's, and show in Fig. 4 (a) the mean errors obtained. It can be seen that the CS method yields reasonable results for σ as low as 0.03, and as high as 0.17, with mean errors less than five obtained in the range $0.05 \leq \sigma \leq 0.14$. Fig. 4 (b) show the result of a similar experiment for $\sigma = 0.09$, $N_{init} = 10$, for a wide range of K_{init}. For $K_{init} < 10$, typically one or more of the ten runs result in complete failure (more than 100 errors) because of the random initialization of the initial clusters. We only show the resulting mean error when there are no complete failures. The error bars indicate the standard deviation. It can be seen that for $K_{init} \geq 17$ the CS method is very stable, and results in a mean error close to three. For $\sigma = 0.09$ we never achieved a better result than three errors. If we choose our clustering result to be the one for which the CS divergence is the smallest among the ten runs for each K_{init}, the result would be three errors for every single K_{init}, even for $K_{init} = 4$. Finally, in Fig. 5 we show the corresponding Parzen pdf estimates for three different σ's. For $\sigma = 0.03$ the pdf estimate is very crude and noisy. For $\sigma = 0.17$ the smoothing effect increasingly dominates. As shown earlier, even for these clearly inaccurate pdf estimates, the CS method performs relatively well. The pdf estimate for σ in the middle range (0.1) is also shown, for which the clustering results naturally are the best.

We also test our method on a data set consisting of highly irregular clusters. For very small σ-values (0.04), up to $\sigma = 0.12$, we obtain a perfect clustering. For example, in Fig. 6 (a) we show the result obtained for $\sigma = 0.1$, $K_{init} = 20$ and $N_{init} = 10$. For $\sigma > 0.12$ the resulting clusters tend to spread over the true cluster boundaries. Not surprisingly, K-means fails completely, as shown in Fig. 6 (b).

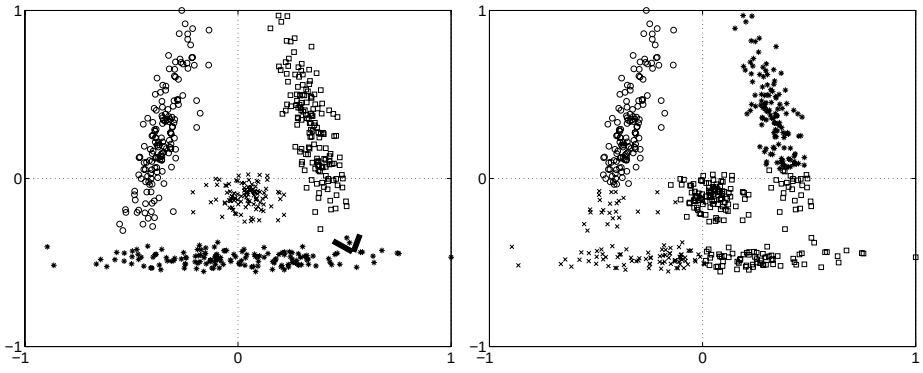
Next, we cluster the well known IRIS[1] data set. This data set consists of three classes of 50 patterns each, where each class refers to a type of iris plant. It is characterized by four numeric attributes. We run the CS algorithm ten times, and select the clustering result yielding the largest CS divergence as our final result. In this case $K_{init} = 10$ and $N_{init} = 10$. For $0.06 \leq \sigma \leq 0.3$ we obtain less than ten errors. The best result is achieved for $\sigma = 0.1$, for which we obtain only five errors. The best clustering result to our knowledge of the IRIS data set is three errors obtained by Roberts et al. [7].

Finally, we test our method on the WINE data set. This data set consists of 178 instances in a 13-dimensional feature space, where the features are found by chemical analysis of three different types of wines. We include this data set in our analysis because it shows that our algorithm is capable of performing well in a high dimensional feature space. For a wide range of σ's we obtain in the order of ten errors. The best result was obtained for $\sigma = 0.5$, yielding six errors.

5. CONCLUSION

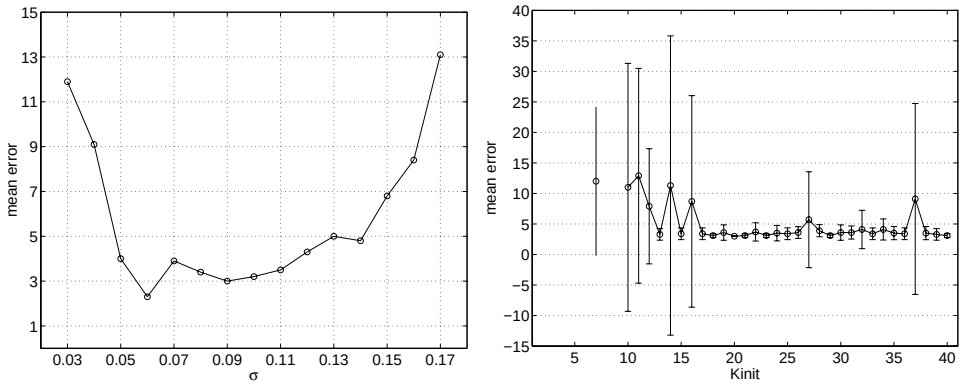
We have introduced the Cauchy-Schwartz pdf divergence measure, and showed that it provides us with

[1]IRIS and WINE data sets extracted from the UCI repository, University of California, Irvine.



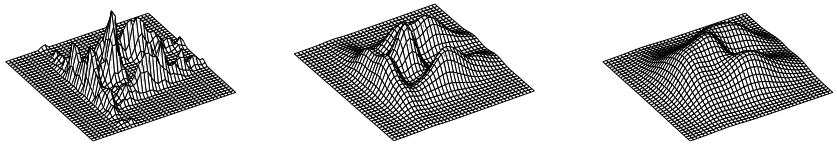
(a) CS-method $\sigma = 0.06$ (b) K-means

Figure 3: (a) Best result for CS-method, and (b) best result using K-means.



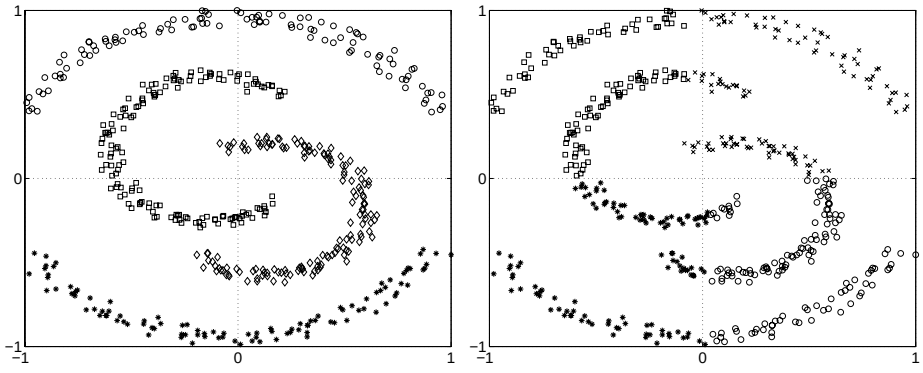
(a) Mean error for different σ. (b) Mean error for different K_{init}

Figure 4: Sensitivity analysis wrt. σ and K_{init}.



(a) $\sigma = 0.03$ (b) $\sigma = 0.1$ (c) $\sigma = 0.17$

Figure 5: Parzen estimate for different σ.



(a) CS-method $\sigma = 0.1$ (b) K-means

Figure 6: (a) Best result for CS-method, and (b) best result using K-means.

a non-parametric information-theoretic cluster evaluation function. The main advantage of our clustering approach is that the underlying clustering metric is based on entropy, both between sub groups, and within sub groups. Entropy is a quantity that conveys information about the shape of probability distributions, and not only variance, which many traditional clustering algorithms, e.g. K-means rely on. This enables us to cluster data sets consisting of elongated and highly irregular clusters.

At present, the major problem with our clustering algorithm is how to choose the kernel size σ. This is a problem encountered in all kernel-based methods, both supervised and unsupervised. However, we have shown that the CS method is not too sensitive to the actual kernel size. Important topics for future research include developing an automatic procedure to determine σ such that the corresponding Parzen pdf estimate is relatively accurate.

6. REFERENCES

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[2] D. Judd, P. McKinley, and A. K. Jain, "Large-Scale Parallel Data Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 871–876, 1998.

[3] H. Frigui and R. Krishuapuram, "A Robust Competitive Clustering Algorithm with Applications in Computer Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 450–465, 1999.

[4] H. M. Abbas and M. M. Fahmy, "Neural Networks for Maximum Likelihood Clustering," *Signal Processing*, vol. 36, no. 1, pp. 111–126, 1994.

[5] C. Carpineto and G. Romano, "A Lattice Conceptual Clustering System and its Application to Browsing Retrieval," *Machine Learning*, vol. 24, no. 2, pp. 96–122, 1996.

[6] J. McQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[7] S. J. Roberts, R. Everson, and I. Rezek, "Minimum Entropy Data Partitioning," in *Ninth International Conference on Artificial Neural Networks*, 1999, vol. 2, pp. 844–849.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & sons, 1991.

[9] S. Watanabe, *Pattern Recognition: Human and Mechanical*, John Wiley & sons, 1985.

[10] R. Jenssen, K. E. Hild, D. Erdogmus, J. C. Principe, and T. Eltoft, "Clustering using Renyi's Entropy," in *International Joint Conference on Neural Networks*, Portland, Oregon, USA, 2003, pp. 523–528.

[11] N. Tishby and N. Slonim, "Data Clustering by Markovian Relaxation and the Information Bottleneck Method," in *Advances in Neural Information Processing Systems, 13*, Denver, USA, 2000, pp. 640–646.

[12] J. Principe, D. Xu, and J. Fisher, "Information Theoretic Learning," in *Unsupervised Adaptive Filtering*, S. Haykin (Ed.), John Wiley & Sons, 2000, vol. I, Chapter 7.

[13] E. Parzen, "On the Estimation of a Probability Density Function and the Mode," *Ann. Math. Stat.*, vol. 32, pp. 1065–1076, 1962.

[14] E. Gokcay and J. Principe, "Information Theoretic Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 158–170, 2002.

[15] M. Girolami, "Mercer Kernel-Based Clustering in Feature Space," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 780–784, 2002.

Information Theoretic Spectral Clustering

Robert Jenssen†, Torbjørn Eltoft† and Jose C. Principe‡

† Electrical Engineering Group, Department of Physics
University of Tromsø, Prestvannvn. 38, N-9011 Tromsø, Norway

‡ Computational NeuroEngineering Laboratory, Department of Electrical and Computer Engineering
University of Florida, Gainesville FL. 32611, USA

***Abstract*— We discuss a new information-theoretic framework for spectral clustering that is founded on the recently introduced *Information Cut*. A novel spectral clustering algorithm is proposed, where the clustering solution is given as a linearly weighted combination of certain top eigenvectors of the data affinity matrix.**

The Information Cut provides us with a theoretically well defined graph-spectral cost function, and also establishes a close link between spectral clustering, and non-parametric density estimation. As a result, a natural criterion for creating the data affinity matrix is provided.

We present preliminary clustering results to illustrate some of the properties of our algorithm, and we also make comparative remarks.

I. Introduction

A new line of research has recently emerged in the context of segmentation and clustering. It is known as the *spectral clustering* methods. These are methods that use the eigenvectors corresponding to certain eigenvalues (the spectrum) of a suitably chosen matrix, $\mathbf{G}$, to partition the data. Usually, $\mathbf{G}$ is the affinity matrix of the data, or a function thereof, such as the closely related Laplacian matrix.

The spectral clustering problem is often cast in graph theoretic terms. A graph consists of the node set C, with a symmetric similarity weight, $G_{ij} \geq 0$, corresponding to the edge between nodes i and j. It is the matrix $\mathbf{G} = [G_{ij}]_{i,j \in C}$ that is called the affinity matrix. A graph can be bi-partitioned into two disjoint sets, or clusters, simply by removing edges between the two parts.

One measure of similarity between these two pieces can be computed as the total weight of the edges that have been removed. This quantity is called *the cut*. The minimum-cut criteria has been used in spectral clustering in [1]. However, the cut-cost is known to favor a skewed partition. To compensate for this fact, a number of rather heuristically motivated improvements to the cut-cost have been proposed. Specifically, we mention the normalized cut [2], the min-max cut [3], the ratio cut (average cut) [4], [5] and the foreground cut [6].

In all these methods, the bi-partitioning is performed based on a single eigenvector of the matrix $\mathbf{G}$. This spectral solution is normally obtained by formulating the cost function as a Rayleigh quotient [7], and by allowing the indicator vector, or membership vector, to take real values. Normally, finding more than two clusters requires a recursive implementation of the method. Procedures for finding multi-way cuts have also been proposed [8]. Other related methods are presented in [9], [10]. For a unifying review of spectral clustering, see [11].

Despite that spectral clustering methods have been observed empirically to work well in a number of cases, the reason for their success is not well understood theoretically. For example, it is often not clear what criteria are optimized by spectral clustering. It is also not clear how to choose an edge-weight function, G_{ij}, such that it reflects the similarity between nodes i and j. In the literature, G_{ij} often depends on user-specified parameters. Automatic procedures for proper selection of these parameters are rarely discussed.

In this paper, we discuss a new framework for spectral clustering based on the recently introduced *Information Cut* (IC) [12]. In theory, clustering based on minimizing the IC leads to a high-dimensional, discrete optimization problem, which is difficult to solve. We express the IC in terms of the eigenvalues and eigenvectors of the data affinity matrix, and approximate the discrete solution by a real-valued solution. This real-valued solution is given as a linear combination of a few top eigenvectors, each weighted by the sum of its own elements. We solve the resulting *low-dimensional* optimization problem by developing a greedy algorithm which is able to select the right eigenvectors. To obtain the final discrete solution, we threshold the real-valued solution. There is no need to search for the best threshold, it is given by our theoretical derivations.

The main appeal of our approach is that it establishes a direct connection between graph-theoretic spectral clustering and the minimization of an information-theoretic distance measure. Furthermore, we show that spectral clustering is closely linked to non-parametric density estimation via the Parzen method, using a Gaussian kernel. As a consequence of this link, our edge-weight function is given by the Gaussian kernel. For optimum clustering performance, the width of the kernel should be chosen such that the pdf estimates are relatively accurate. This is a problem in itself, but a theoretical criterion for the selection of the kernel size may be formulated in pdf estimation.

We compare our clustering result to the clustering obtained by the foreground cut algorithm [6], since this method uses only the eigenvector corresponding to the largest eigenvalue

This work was partially supported by NSF grants ECS-9900394 and EIA 0135946.

of the affinity matrix in its solution. We show that this is in general not sufficient.

The organization of this paper is as follows. In section II we define the Information Cut. In section III we derive a spectral clustering algorithm based on minimizing the IC. Thereafter, in section IV, we perform some comparative clustering experiments. We give our concluding remarks in section V.

II. The Information Cut

Recently, Principe et al. [13] proposed a new information-theoretic pdf distance measure based on the Cauchy-Schwarz (CS) inequality between two vectors. The reason for introducing this new distance measure was that it elegantly integrates non-parametric pdf estimation through Parzen windowing. This means that we can evaluate the distance measure between pdfs, without making any parametric assumptions about the underlying distributions.

Based on the Cauchy-Schwarz inequality; $||\mathbf{x}||^2\ ||\mathbf{y}||^2 \geq (\mathbf{x}^T\mathbf{y})^2$, we may write;

$$-\log \frac{\mathbf{x}^T\mathbf{y}}{\sqrt{||\mathbf{x}||^2\ ||\mathbf{y}||^2}} \geq 0. \tag{1}$$

By replacing inner products between vectors in (1), by inner products between pdfs, i.e. $\langle p, q\rangle = \int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}$, we define the CS distance [13];

$$D_{CS} = -\log \frac{\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}}{\sqrt{\int p^2(\mathbf{x})d\mathbf{x}\ \int q^2(\mathbf{x})d\mathbf{x}}} \geq 0. \tag{2}$$

In order for D_{CS} to equal zero, the two pdfs must overlap completely. It goes to infinity as the overlap between the two pdfs goes to zero.

Assume that we estimate $p(\mathbf{x})$ based on the data points in cluster $C_1 = \{\mathbf{x}_i\},\ i = 1, \ldots, N_p$, and $q(\mathbf{x})$ based on $C_2 = \{\mathbf{x}_j\},\ j = 1, \ldots, N_q$. By the Parzen [14] method;

$$\begin{aligned}\hat{p}(\mathbf{x}) &= \frac{1}{N_p}\sum_{i=1}^{N_p} G(\mathbf{x} - \mathbf{x}_i, \sigma^2\mathbf{I}),\\ \hat{q}(\mathbf{x}) &= \frac{1}{N_q}\sum_{j=1}^{N_q} G(\mathbf{x} - \mathbf{x}_j, \sigma^2\mathbf{I}),\end{aligned} \tag{3}$$

where we have used the multi-dimensional Gaussian kernel, $G(\mathbf{x}, \mathbf{\Sigma})$, and $\mathbf{\Sigma} = \sigma^2\mathbf{I}$. Since maximization of D_{CS} is equivalent to minimization of the argument of the logarithm in (2), we now derive the expression for the latter quantity, which we denote the Information Cut (IC). This is done by substituting the pdfs by their Parzen estimates, and by utilizing the convolution theorem for Gaussians, which states that;

$$\int G(\mathbf{x} - \mathbf{x}_i, \sigma^2\mathbf{I})G(\mathbf{x} - \mathbf{x}_j, \sigma^2\mathbf{I})d\mathbf{x} = G_{ij,2\sigma^2\mathbf{I}}, \tag{4}$$

where $G_{ij,2\sigma^2\mathbf{I}} = G(\mathbf{x}_i - \mathbf{x}_j, 2\sigma^2\mathbf{I})$. Thus, when we plug the Parzen pdf estimates of (3) into the argument of (2), and utilize (4), we obtain;

$$\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x} \approx \frac{1}{N_pN_q}\sum_{i,j=1}^{N_p,N_q} G_{ij,2\sigma^2\mathbf{I}}, \tag{5}$$

$$\int p^2(\mathbf{x})d\mathbf{x} \approx \frac{1}{N_P^2}\sum_{i,i'=1}^{N_p,N_p} G_{ii',2\sigma^2\mathbf{I}}, \tag{6}$$

and likewise for $\int q^2(\mathbf{x})d\mathbf{x}$, such that;

$$IC = \frac{\sum_{i,j=1}^{N_p,N_q} G_{ij,2\sigma^2\mathbf{I}}}{\sqrt{\sum_{i,i'=1}^{N_p,N_p} G_{ii',2\sigma^2\mathbf{I}}\ \sum_{j,j'=1}^{N_q,N_q} G_{jj',2\sigma^2\mathbf{I}}}}. \tag{7}$$

If $\mathbf{x}_i$ and $\mathbf{x}_j$ are considered nodes in the set C, it is clear that $G_{ij,2\sigma^2\mathbf{I}}$ is in fact the similarity weight between the two nodes. This makes explicit the link between the edge-weight function and pdf estimation. It can be seen that the numerator of the Information Cut is exactly the traditional *cut* known from graph theory. The denominator of (7) acts as a normalizing quantity, which in our experience helps avoid the problems associated with the cut-cost alone with regard to a skewed partition. It is not motivated by heuristics, but derived from the information-theoretic origin.

Equation (7) can be written in a more compact form. We define the affinity matrix $\mathbf{G} = \left[G_{ij,2\sigma^2\mathbf{I}}\right]_{i,j\in C}$. Furthermore, we define an N-dimensional indicator vector, or membership vector, $\mathbf{m}$, where $N = N_p + N_q = |C|$, such that $m_i = 1$ if node i is in C_1 and 0, if it is in C_2. Hence we obtain;

$$IC = \frac{\mathbf{m}^T\mathbf{G}(\mathbf{1} - \mathbf{m})}{\sqrt{\mathbf{m}^T\mathbf{G}\mathbf{m}(\mathbf{1} - \mathbf{m})^T\mathbf{G}(\mathbf{1} - \mathbf{m})}}. \tag{8}$$

In order to perform a two-way clustering of a dataset, the goal is to determine $\mathbf{m}$ such that the value of the Information Cut is minimized, because this corresponds to maximizing the CS distance between the two clusters. However, $\mathbf{m}$ is a binary vector, so we are faced with a discrete minimization task, which is very difficult to solve efficiently. We take the common approach of approximating the discrete solution by allowing $\mathbf{m}$ to take analog values, and to impose certain constraints on the solution.

III. Spectral clustering solution

Symmetry guarantees that all of $\mathbf{G}$'s eigenvalues are real and that there is an orthonormal basis of eigenvectors. By the Schur decomposition [7], $\mathbf{G}$ can be written as;

$$\mathbf{G} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^T = \mathbf{E}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{E}^T, \tag{9}$$

where the columns of $\mathbf{E}$ contain the eigenvectors, and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$ is a diagonal matrix that contains the corresponding eigenvalues in decreasing order.

Now we define the vector

$$\mathbf{u} = \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{E}^T\mathbf{m}. \tag{10}$$

It follows that;

$$\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{E}^T(\mathbf{1} - \mathbf{m}) = \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{E}^T\mathbf{1} - \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{E}^T\mathbf{m} = \mathbf{t} - \mathbf{u}, \tag{11}$$

where $\mathbf{t} = \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{E}^T\mathbf{1}$ and $\mathbf{1}$ is a $(N \times 1)$ vector whose elements are all one. By inserting (9) into (8), and utilizing (10) and (11), we obtain the following expression for the IC;

$$IC = \frac{\mathbf{u}^T(\mathbf{t} - \mathbf{u})}{\sqrt{||\mathbf{u}||^2||\mathbf{t} - \mathbf{u}||^2}} = \cos\angle(\mathbf{u}, \mathbf{t} - \mathbf{u}). \tag{12}$$

We proceed by seeking the $\mathbf{u}$ that satisfies;

$$\min_{\mathbf{u}} \ IC = \min_{\mathbf{u}} \ \cos \angle(\mathbf{u}, \mathbf{t} - \mathbf{u}). \tag{13}$$

In order to be able to utilize (13) we need to obtain some constraints on $\mathbf{u}$. We know that for any discrete solution, the following holds: $\mathbf{m}^T(\mathbf{1} - \mathbf{m}) = 0$. This is equivalent to the following constraint on $\mathbf{u}$;

$$\mathbf{u}^T \mathbf{\Lambda}^{-1}(\mathbf{t} - \mathbf{u}) = 0, \tag{14}$$

which in fact is the equation of a hyper-ellipse in the N-dimensional space. Hence, we seek the $\mathbf{u}$ that lies on the hyper-ellipse described by (14), and at the same time minimizes the IC.

Now, notice that any vector $\mathbf{u}$ of the form;

$$u_i \in \{t_i, 0\}, \ i = 1, \ldots, N, \tag{15}$$

does indeed obey (14). Furthermore, it has the property that;

$$\mathbf{u} \perp \mathbf{t} - \mathbf{u} \Rightarrow \cos \angle(\mathbf{u}, \mathbf{t} - \mathbf{u}) = 0, \tag{16}$$

which means that such a vector minimizes the Information Cut.

Remember that $\mathbf{u} = \mathbf{\Lambda}^{\frac{1}{2}} \mathbf{E}^T \mathbf{m}$, such that;

$$\mathbf{m} = \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{E} \mathbf{u} = \sum_{i=1}^{N} \frac{u_i}{\sqrt{\lambda_i}} \ \mathbf{e}_i, \tag{17}$$

where $\mathbf{e}_i$ is the i'th column of $\mathbf{E}$, i.e. the i'th eigenvector of $\mathbf{G}$. Define $w_i = 1$ if $u_i = t_i$, and $w_i = 0$ if $u_i = 0$, $i = 1, \ldots, N$. Thus, the discrete solution can be written;

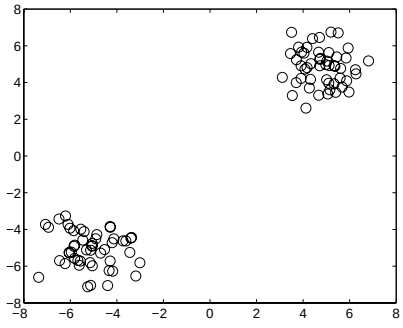
$$\mathbf{m} = \sum_{i=1}^{N} w_i \frac{t_i}{\sqrt{\lambda_i}} \ \mathbf{e}_i = \sum_{i=1}^{N} w_i (\mathbf{e}_i^T \mathbf{1}) \ \mathbf{e}_i, \tag{18}$$

since $t_i = \sqrt{\lambda_i} \ \mathbf{e}_i^T \mathbf{1}$. In conclusion, our solution is given as a linearly weighted summation of some of the eigenvectors, where the weighting on each eigenvector is given by the sum of the elements of that eigenvector.

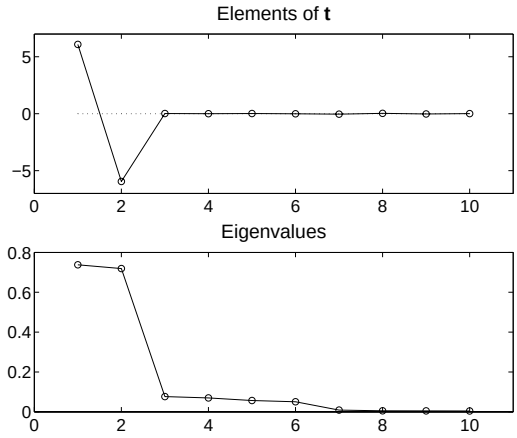
Now we need to determine which eigenvectors to use, such that (18) is indeed the correct discrete solution. Of course, $\mathbf{u} = \mathbf{0}$ or $\mathbf{u} = \mathbf{t}$ are not valid solutions, since these solutions translate into $\mathbf{m} = \mathbf{0}$ or $\mathbf{m} = \mathbf{1}$. These are trivial solutions, meaning that all data points are assigned to only one cluster.

A. The real-valued solution: reducing complexity

We are faced with the problem of finding some N-dimensional vector $\mathbf{u}$ of the form (15). Any such vector will minimize the IC, which makes this a high-dimensional optimization problem. Fortunately, it turns out that most of the elements of the vector $\mathbf{t}$ are typically very close to zero, a fact that we have observed experimentally on a number of datasets. There are in fact only a small number M of such elements, where $M \ll N$, deviating significantly from zero. We embed our optimization problem in the M-dimensional real-value domain, by relaxing the discrete vector $\mathbf{m}$ to take analog values, and proceed in order to determine a vector $\hat{\mathbf{u}}$, such that $\hat{u}_i = t_i$ or $\hat{u}_i = 0$, for $i = 1, \ldots, M$. Hence $\hat{\mathbf{m}} = \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{E} \hat{\mathbf{u}}$.



(a)



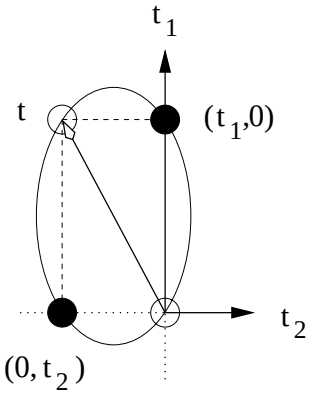
(b)

Fig. 1. Dataset consisting of two Gaussian clusters, shown in (a). Elements of $\mathbf{t}$ and $\mathbf{\Lambda}$ are shown in (b).

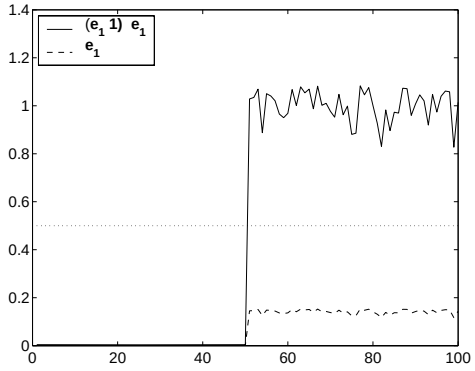
To obtain the discrete solution, $\mathbf{m}$, we threshold $\hat{\mathbf{m}}$ such that elements larger than $1/2$ are given the value one, and elements smaller than $1/2$ are given the value zero.

Of course, the number of significantly large elements of $\mathbf{t}$ varies from dataset to dataset, and is also closely dependent on the edge-weight function used, in our case it depends on the parameter σ. In our experience, one reason for this behaviour of $\mathbf{t}$ is that there are typically only a few of the eigenvalues of the affinity matrix that deviate significantly from zero. Since $t_i = \sqrt{\lambda_i} \ \mathbf{e}_i^T \mathbf{1}$, it will be close to zero if λ_i is close to zero. This is true since $\mathbf{E}$ is orthonormal, hence the sum of the elements of each eigenvector is bounded, since $\mathbf{e}_i^T \mathbf{e}_i = 1$. Another reason is that frequently $\mathbf{e}_i^T \mathbf{1} \approx 0$, such that t_i may also be very close to zero even if λ_i is not.

As an example, consider the simple dataset shown in Fig. 1 (a). For a suitably chosen σ, only two of the elements of $\mathbf{t}$ deviate significantly from zero. This is illustrated in the upper panel of Fig. 1 (b), where the ten first entries of $\mathbf{t}$ have been shown for $\sigma = 2.14$. We will return to the selection of σ in section III-C. In fact, for this dataset, there are also only two



(a)



(b)

Fig. 2. The solution ellipse (14) is shown in (a). The real-valued solution $\hat{\mathbf{m}}$ and the largest eigenvector $\mathbf{e}_1$ is shown in (b). The horizontal line (dotted) indicates the threshold at $1/2$.

significant eigenvalues. This is illustrated in the lower panel of Fig. 1 (b), where the first ten entries on the diagonal of $\mathbf{\Lambda}$ have been shown.

Fig. 2 (a) shows the ellipse described by (14) for $\mathbf{t} = [t_1, t_2]^T$. The IC is minimized by selecting either $\hat{\mathbf{u}} = [t_1, 0]^T$ or $\hat{\mathbf{u}} = [0, t_2]^T$, since it follows that $\hat{\mathbf{u}} \perp \mathbf{t} - \hat{\mathbf{u}}$. These two solutions are equivalent; they both divide the graph into the same two groups, but they give opposite labels to the two clusters. Let us examine the solution based on $\hat{\mathbf{u}} = [t_1, 0]^T$, that is, $\hat{\mathbf{m}} = (\mathbf{e}_1^T \mathbf{1})\ \mathbf{e}_1$. In Fig. 2 (b) we show a plot of both $\hat{\mathbf{m}}$ and $\mathbf{e}_1$. The dataset has been ordered, such that the first 50 elements of these vectors correspond to the cluster to the left, and the last 50 correspond to the other cluster. Clearly, $\hat{\mathbf{m}}$ approximates the discrete solution very closely. We conclude that our information-theoretic approach has revealed that when there are two significant elements of $\mathbf{t}$, then there are two equivalent solutions, namely either $\hat{\mathbf{m}} = (\mathbf{e}_1^T \mathbf{1})\ \mathbf{e}_1$ or $\hat{\mathbf{m}} = (\mathbf{e}_2^T \mathbf{1})\ \mathbf{e}_2$.

Note that the foreground cut algorithm [6] uses the eigenvector corresponding to the largest eigenvalue to partition the graph. Hence, for a dataset such as the one described above, we expect our algorithm and the foreground cut algorithm to obtain similar clustering results. However, when using the foreground cut, it is not clear how to choose the threshold. Fig. 2 (b) clearly shows that in this case there exists a threshold that yields the same result as that given by the IC.

In general, however, there is no guarantee that there will be only two significant elements of $\mathbf{t}$. In such cases the foreground cut algorithm will fail.

In the following we briefly describe our new information theoretic spectral clustering algorithm.

B. The spectral algorithm

The main idea behind our algorithm is very simple. We take as our starting point that the eigenvector $\mathbf{e}_1$, corresponding to t_1, is a part of the sum (18) that constitutes the solution. One-by-one, we include temporarily the other eigenvectors corresponding to the remaining $t_i,\ i = 2, \ldots, M$, into the sum. In each case the IC is calculated. We pick the next eigenvector to be appended permanently, to be the component that yields the smallest IC-value. This procedure is terminated if the IC-value increases from one iteration to the next. A pseudo-code for this algorithm is given below. We denote by $\mathbf{e}'$ an eigenvector that has not yet been included in the solution.

- `Eigendecompose` $\mathbf{G}$
- `Calculate` $\mathbf{t}$
- $M =$ `# significant` t_i`'s`
- `Initialize:` $\hat{\mathbf{m}} = (\mathbf{e}_1^T \mathbf{1})\ \mathbf{e}_1$
- `Determine` $IC(\mathbf{m})$
- `while loop`
  - `for j = 1 : M-1`
    $\hat{\mathbf{m}}_j = \hat{\mathbf{m}} + (\mathbf{e}_j'^T \mathbf{1})\ \mathbf{e}_j'$
    `Store` $IC(\mathbf{m}_j)$
    `end for`
  - `Find` $\hat{\mathbf{m}}_{\min} : \min_{\hat{\mathbf{m}}_j} IC(\mathbf{m}_j),\ \ j = 1, \ldots, M-1$
  - `if` $IC(\mathbf{m}_{\min}) < IC(\mathbf{m})$
    $\hat{\mathbf{m}} = \hat{\mathbf{m}}_{\min}$
    $M = M - 1$
    `else stop`
    `end if`

`until stop`

The number M of significant elements of $\mathbf{t}$ is determined by selecting the elements whose value is larger than $0.1 \times t_{i,\max},\ i = 1, \ldots, N$. In order to cluster the data into several groups, we apply this spectral algorithm recursively.

C. Creating the affinity matrix

A major drawback of the vast majority of the existing spectral clustering algorithms, is that it is not clear how to choose an edge-weight function in order to construct the affinity matrix. In many cases, the Gaussian edge-weight function is used, which requires the user to specify the width, σ, of the Gaussian kernel. It has been observed that the results obtained by various algorithms may vary significantly depending on σ. This also makes a fair comparison between different algorithms difficult.

Using the Information Cut, σ is in fact closely tied to Parzen pdf estimation. Selecting a proper kernel size for Parzen pdf estimation is by no means a trivial task, especially for high-dimensional data. Still, we will show that we are able to obtain promising preliminary clustering results using our Information Cut algorithm, where the affinity matrix is automatically created. Here, we briefly state Silverman's [15] "rule-of-thumb" for automatic selection of the "optimal" kernel size for Parzen pdf estimation.

The mean integrated square error (MISE) is the most widely used measure on the global accuracy of an estimator $\hat{p}(\mathbf{x})$ of the pdf $p(\mathbf{x})$;

$$MISE(\hat{p}) = E \int \{\hat{p}(\mathbf{x}) - p(\mathbf{x})\}^2 d\mathbf{x}. \tag{19}$$

The MISE can be approximated asymptotically, from which an expression for the optimal kernel size, σ_{opt}, can be derived. The problem is that σ_{opt} itself depends on the second derivative of the density we wish to estimate. The "rule-of-thumb" approach is to obtain a rough estimate of the unknown quantity by assuming that the underlying density is Gaussian. This rough estimate is then plugged back into the expression for σ_{opt}. Hence, σ_{opt} will only be optimal for a Gaussian density, and will normally result in a over-smoothed estimate for non-Gaussian data.

For d-dimensional data, the "optimal" kernel size is given by [15];

$$\sigma_{\text{opt}} = s \left(\frac{4}{N(2d+1)} \right)^{\frac{1}{d+4}}, \tag{20}$$

where $s^2 = d^{-1} \sum_i S_{ii}$, and S_{ii} are the diagonal elements of the sample covariance matrix. In our experiments, we have utilized (20) in order to create the affinity matrix. However, for strongly non-Gaussian datasets it has a tendency to over-smooth. We therefore suggest to estimate the optimal one-dimensional kernel size for each dimension of the data, and use the smallest such value as our σ. The one-dimensional "optimal" bandwidth is given by [15];

$$\sigma_{\text{opt}} = s 1.06 N^{-1/5}, \tag{21}$$

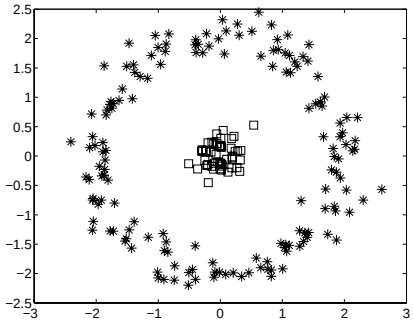
where s is an estimate of the standard deviation of the one-dimensional data.

In all examples and clustering experiments in this paper, the data affinity matrix is automatically created based on (21) and the above discussion.

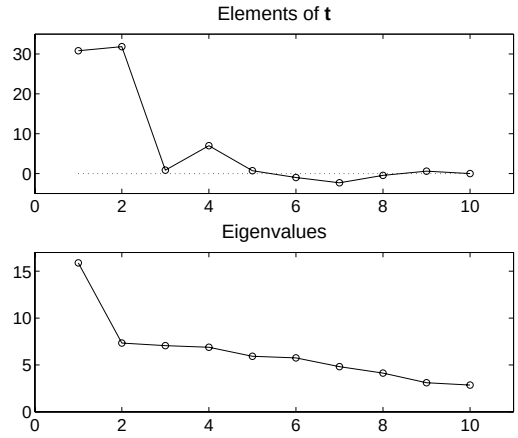
IV. Clustering Experiments

In the first experiment we consider a dataset consisting of a concentric ring with a Gaussian cluster in the middle. We provide the dataset only to our algorithm, there are no user-specified parameters. The output is shown in Fig. 3 (a), which is in fact the correct solution.

In Fig. 3 (b), the first ten entries of $\mathbf{t}$, and of the diagonal of $\mathbf{\Lambda}$, are shown in the upper and lower panels, respectively. The algorithm selects three significant elements of $\mathbf{t}$, corresponding to the eigenvalues λ_1, λ_2 and λ_4. Note that all displayed eigenvalues have fairly large values, as opposed to the previous example.



(a)



(b)

Fig. 3. Dataset consisting of a concentric ring with a Gaussian cluster in the middle. The correct clustering is shown in (a), while elements of $\mathbf{t}$ and $\mathbf{\Lambda}$ are shown in (b).

The algorithm determines the solution to be $\hat{\mathbf{m}} = (\mathbf{e}_1 \mathbf{1})\ \mathbf{e}_1$, and the opposite solution (switching labels) to be given by $(\mathbf{e}_2 \mathbf{1})\ \mathbf{e}_2 + (\mathbf{e}_4 \mathbf{1})\ \mathbf{e}_4$. Hence, also in this case the foreground cut algorithm should work well since a solution based on the largest eigenvector can be found. This is indeed the case, using the same σ as our algorithm determined.

In the second experiment we consider a dataset consisting of two equally sized half-circles. The output of the algorithm is shown in Fig. 4 (a). The clustering is completely correct.

Fig. 4 (b) shows that there are seven significant elements of $\mathbf{t}$. All displayed eigenvalues are large, but decreasing slowly in value. The algorithm determines that clustering based on the largest eigenvector alone is not sufficient for this dataset. It includes another two eigenvectors in the sum (18).

Fig. 4 (c) shows a plot of the largest eigenvector, $\mathbf{e}_1$. Also in this case the dataset has been ordered, such that the first half of the elements of $\mathbf{e}_1$ correspond to the cluster to the left, and the second half to the cluster to the right. There is no way to find a threshold resulting in a perfect clustering. In this case,

the foreground cut algorithm fails.

V. Conclusions

We have discussed a new information-theoretic framework for spectral clustering based on the Information Cut.

We have developed a novel spectral clustering algorithm, where the real-valued clustering solution is given as a linearly weighted combination of a few of the top eigenvectors, where the weighting on each eigenvector is given by the sum of the elements of that vector. This algorithm is fully automatic, since there are no user-specified parameters needed in order to construct the affinity matrix. This is a major advantage compared to other spectral methods in the literature.

Another advantage is that our spectral clustering cost function is well defined, originating from an information-theoretic pdf distance measure based on the Cauchy-Schwarz inequality. Thus, an interesting link between graph-theory and information-theory is revealed, which also shows that spectral clustering is closely related to non-parametric pdf estimation.

In our future work, we will further study the issue of automatic selection of a proper kernel size for relatively accurate Parzen density estimation. This may be of special importance when it comes to high-dimensional data.

References

[1] N. Cristianini, J. Shawe-Taylor, and J. Kandola, "Spectral Kernel Methods for Clustering," in *Advances in Neural Information Processing Systems, 14*, 2001, vol. 1, pp. 649–655.

[2] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[3] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A Min-max Cut Algorithm for Graph Partitioning and Data Clustering," in *IEEE Int. Conf. on Data Mining*, 2001, pp. 107–114.

[4] L. Hagen and A. B. Kahng, "Fast Spectral Methods for Ratio Cut Partitioning and Clustering," in *International Conference on Computer-Aided Design*, Santa Clara, CA, USA, 1991, pp. 10–13.

[5] S. Sarkar and P. Soundararajan, "Supervised Learning of Large Perceptual Organization: Graph Spectral Partitioning and Learning Automata," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 5, pp. 504–525, 2000.

[6] P. Perona and W. T. Freeman, "A Factorization Approach to Grouping," in *Proc. European Conference on Computer Vision*, 1998, pp. 655–670.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The John Hopkins University Press, 1996.

[8] M. Meila and L. Xu, "Multiway Cuts and Spectral Clustering," Tech. Rep. 442, University of Washington, Deartment of Statistics, January 2004.

[9] A. Y. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," in *Advances in Neural Information Processing Systems, 14*, 2001, vol. 2, pp. 849–856.

[10] R. Kannan, S. Vempala, and A. Vetta, "On Clusterings: Good, Bad and Spectral," in *IEEE Foundations of Computer Science*, Redondo Beach, CA, USA, 2000, pp. 367–377.

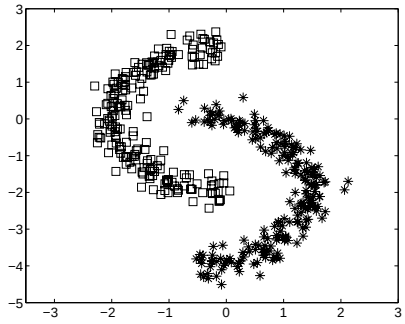
[11] Y. Weiss, "Segmentation Using Eigenvectors: A Unifying View," in *International Conference on Computer Vision*, 1999, pp. 975–982.

[12] R. Jenssen, J. C. Principe, and T. Eltoft, "Information Cut and Information Forces for Clustering," in *IEEE International Workshop on Neural Networks for Signal Processing*, Toulouse, France, 2003, pp. 459–468.

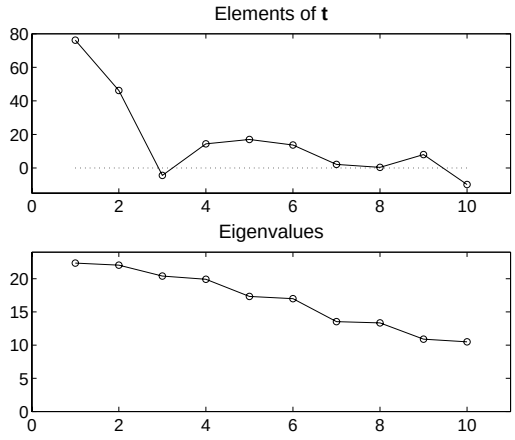
[13] J. Principe, D. Xu, and J. Fisher, "Information Theoretic Learning," in *Unsupervised Adaptive Filtering*, S. Haykin (Ed.), John Wiley & Sons, 2000, vol. I, Chapter 7.

[14] E. Parzen, "On the Estimation of a Probability Density Function and the Mode," *Ann. Math. Stat.*, vol. 32, pp. 1065–1076, 1962.

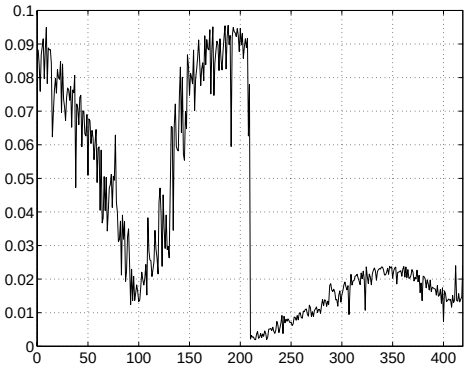
[15] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, 1986.



(a)



(b)



(c)

Fig. 4. Dataset consisting of two half-circles. The correct clustering is shown in (a), while elements of $\mathbf{t}$ and $\mathbf{\Lambda}$ are shown in (b).

Bibliography

J. Aczél and Z. Daróczy. *On Measures of Information and Their Characterizations*. Academic Press, New York, 1975.

C. Alpert and S. Yao. Spectral Partitioning: The More Eigenvectors the Better. In *Proceedings of ACM/IEEE Design Automation Conference*, San Francisco, USA, June 12-16, 1995.

S. Amari, A. Cichocki, and H. Yang. A New Learning Algorithm for Blind Source Separation. In *Advances in Neural Information Processing Systems, 8*, pages 757–763. MIT Press, Cambridge, 1996.

S.-I. Amari. Neural Learning in Structured Parameter Spaces - Natural Riemannian Gradient. In *Advances in Neural Information Processing Systems, 9*, pages 127–133, MIT Press, Cambridge, 1997.

M. R. Anderberg. *Cluster Analysis for Applications*. Academic Press, New York, 1973.

M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: Ordering Points to Identify the Clustering Structure. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 49–60, Philadelphia, USA, June 1-3, 1999.

J. A. Arrowood and M. A. Clements. Extended Cluster Information Vector Quantization (ECI-VQ) for Robust Classification. In *Proceedings of IEEE International Conference Acoustics, Speech and Signal Processing*, volume 1, pages 889–892, Montreal, Canada, May 17-21, 2004.

J. J. Atick. Could Information Theory Provide an Ecological Theory of Sensory Processing? *Network*, 3:213–251, 1992.

Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia. Spectral Analysis of Data. In *Proceedings of ACM Symposium on Theory of Computing*, pages 619–626, Heraklion, Greece, June 6-8, 2001.

G. H. Ball and D. J. Hall. ISODATA, a Novel Method of Data Analysis and Classification. Technical report, Stanford University, Stanford, USA, 1965.

A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh. Clustering with Bregman Divergences. In *Proceedings of SIAM International Conference on Data Mining*, pages 234–245, Lake Buena Vista, USA, April 22-24, 2004.

J. D. Banfield and A. E. Raftery. Model-Based Gaussian and Non-Gaussian Clustering. *Biometrics*, 49:803–821, 1993.

H. B. Barlow. Unsupervised Learning. *Neural Computation*, 1:295–311, 1989.

H. B. Barlow, T. Kaushal, and G. Mitchison. Finding Minimum Entropy Codes. *Neural Computation*, 1(3):412–423, 1989.

M. Belkin and P. Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15:1373–1396, 2003.

A. J. Bell and T. J. Sejnowski. An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7:1129–1159, 1995.

A. J. Bell and T. J. Sejnowski. The "Independent Components" of Natural Scenes are Edge Filters. *Vision Research*, 37:3327–3338, 1997.

Y. Bengio, J.-F. Paiement, and P. Vincent. Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps and Spectral Clustering. In *Advances in Neural Processing Systems, 16*, MIT Press, Cambridge, 2004.

Y. Bengio, P. Vincent, and J.-F. Paiement. Spectral Clustering and Kernel PCA are Learning Eigenfunctions. Technical report, Département d'informatique et recherche opérationelle, université de Montréal, Montréal, Canada, 2003.

J. C. Bezdek. A Convergence Theorem for the Fuzzy Isodata Clustering Algorithms. *IEEE Transactions on Pattern Analysis and Machine Learning*, 2(1):1–8, 1980.

J. C. Bezdek, M. R. Pal, J. Keller, and R. Krisnapuram. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Kluwer Academic Publishers, Norwell, USA, 1999.

A. Bhattacharyya. On a Measure of Divergence Between Two Statistical Populations Defined by their Probability Distributions. *Bull. Calcutta Math.*, 35:99–109, 1943.

R. Boscolo, H. Pan, and V. P. Royschowdhury. Independent Component Analysis Based on Nonparametric Density Estimation. *IEEE Transactions on Neural Networks*, 15(1):55–65, 2004.

M. Brand. Minimax Embeddings. In *Advances in Neural Information Processing Systems, 16*, MIT Press, Cambridge, 2004.

S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30:107–117, 1998.

C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Knowledge Discovery and Data Mining*, 2(2):121–167, 1998.

R. L. Cannon, J. V. Dave, and J. C. Bezdek. Efficient Implementation of the Fuzzy C-Means Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):248–255, 1986.

J.-F. Cardoso. Blind Identification of Independent Signals. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pages 2109–2112, Glasgow, Scotland, May 23-26, 1989.

J.-F. Cardoso. Eigenstructure of the Fourth-Order Cumulant Tensor with Application to the Blind Source Separation Problem. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pages 2655–2658, Albuquerque, USA, April 3-6, 1990.

J.-F. Cardoso. Infomax and Maximum Likelihood for Source Separation. *IEEE Letters on Signal Processing*, 4:112–114, 1997.

J.-F. Cardoso, C. Jutten, and P. Loubaton, editors. *Proceedings of the First International Workshop on Independent Component Analysis and Blind Signal Separation*, Aussois, France, January 11-15, 1999.

J.-F. Cardoso and A. Souloumiac. Blind Beamforming for Non-Gaussian Signals. *IEE Proceedings-F*, 140(6):362–370, 1993.

G. Carpenter and S. Grossberg. ART3: Hierarchical Search using Chemical Transmitters in Self-Organizing Pattern Recognition Architectures. *Neural Networks*, 3:129–152, 1990.

P. Chang, D. Schlag, and J. Zien. Spectral K-Way Ratio-Cut Partitioning and Clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(9):1088–1096, 1994.

H. Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on a Sum of Observations. *The Annals of Mathematical Statistics*, 23:493–507, 1952.

S. Choi and T.-W. Lee. A Negentropy Minimization Approach to Adaptive Equalization for Digital Communications Systems. *IEEE Transactions on Neural Networks*, 15(4):928–936, 2004.

A. Cichocki and R. Unbehauen. Robust Neural Networks with On-Line Learning for Blind Identification and Blind Separation of Sources. *IEEE Transactions on Circuits and Systems*, 43(11):894–906, 1996.

A. Cichocki, R. Unbehauen, and E. Rummert. Robust Learning Algorithm for Blind Separation of Sources. *Electronics Letters*, 30(17):1386–1387, 1994.

P. Comon. Independent Component Analysis - A New Concept? *Signal Processing*, 36:287–314, 1994.

P. Comon, C. Jutten, and J. Hérault. Blind Separation of Sources, Part II: Problems Statement. *Signal Processing*, 24:11–20, 1996.

P. Comon and B. Mourrain. Decomposition of Quantics in Sums of Powers of Linear Forms. *Signal Processing*, 53(2):93–107, 1996.

C. Cortez and V. N. Vapnik. Support Vector Networks. *Machine Learning*, 20:273–297, 1995.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, 1991.

R. Cristescu, J. Joutsensalo, J. Karhunen, and E. Oja. A Complexity Minimization Approach for Estimating Fading Channels in CDMA Communications. In *Proceedings of International Workshop on Independent Component Analysis and Blind Signal Separation*, pages 527–532, Helsinki, Finland, June 3-6, 2000a.

R. Cristescu, T. Ristaniemi, J. Joutsensalo, and J. Karhunen. Blind Separation of Convolved Mixtures for CDMA Systems. In *Proceedings of European Signal Processing Conference*, pages 619–622, Tampere, Finland, September 5-8, 2000b.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, 2000.

S. A. Cruces-Alvarez, A. Cichocki, and S.-I. Amari. From Blind Source Signal Extraction to Blind Instantaneous Signal Separation: Criteria, Algorithms, and Stability. *IEEE Transactions on Neural Networks*, 15(4):859–873, 2004.

N. Delfosse and P. Loubaton. Adaptive Blind Separation of Independent Sources: A Deflation Approach. *Signal Processing*, 45:59–83, 1995.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, B*, 39(1):1–38, 1977.

L. Devroye. On Random Variate Generation when only Moments or Fourier Coefficients are Known. *Mathematics and Computers in Simulation*, 31:71–89, 1989.

I. S. Dhillon, Y. Guan, and B. Kulis. Kernel K-means, Spectral Clustering and Normalized Cuts. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 551–556, Seattle, USA, August 22-25, 2004.

I. S. Dhillon, J. Kogan, and C. Nicholas. Feature Selection and Document Clustering. In *A Comprehensive Survey of Text Mining*, M. W. Berry (Ed.), Springer Verlag, 2003a.

I. S. Dhillon, S. Maella, and R. Kumar. Enhanced Word Clustering for Hierarchical Text Classification. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 191–200, Edmonton, Canada, July 23-26, 2002.

I. S. Dhillon, S. Maella, and R. Kumar. A Divisive Information-Theoretic Feature Clustering Algorithm for Text Classification. *Journal of Machine Learning Research*, 3:1265–1287, 2003b.

I. S. Dhillon, S. Maella, and R. Kumar. Information-Theoretic Co-Clustering. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 89–98, Washington DC, USA, August 24-27, 2003c.

I. S. Dhillon and D. S. Modha. Concept Decompositions for Large Sparse Text Data using Clustering. *Machine Learning*, 42(1):143–175, 2001.

E. Diday. The Dynamic Cluster Method in Non-Hierarchical Clustering. *Journal of Computer and Information Sciences*, 2:61–88, 1973.

C. H. Q. Ding and X. He. K-Means Clustering via Principal Component Analysis. In *Proceedings of International Conference on Machine Learning*, pages 225–232, Banff, Canada, July 4-8, 2004.

C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A Min-max Cut Algorithm for Graph Partitioning and Data Clustering. In *Proceedings of IEEE International Conference on Data Mining*, pages 107–114, San Jose, USA, November 29 - December 2, 2001.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 2nd edition, 2001.

R. El-Yaniv and O. Souroujon. Iterative Double Clustering for Unsupervised and Semi-Supervised Learning. In *Advances in Neural Information Processing Systems, 14*, pages 1025–1032, MIT Press, Cambridge, 2001.

T. Eltoft and R. J. P. deFigueiredo. A New Neural Network for Cluster-Detection-and-Labeling. *IEEE Transactions on Neural Networks*, 9(5):1021–1035, 1998.

T. Eltoft and Ø. Kristensen. ICA and Nonlinear Time Series Prediction for Recovering Missing Data Segments in Multivariate Signals. In *Proceedings of International Workshop on Independent Component Analysis and Blind Signal Separation*, pages 716–721, San Diego, USA, December 9-12, 2001.

D. Erdogmus. *Information Theoretic Learning: Renyi's Entropy and its Applications to Adaptive Systems Training*. PhD thesis, University of Florida, Gainesville, FL, USA, 2002.

D. Erdogmus, R. Agrawal, and J. C. Principe. A Mutual Information Extension to the Matched Filter. *Signal Processing, to appear*, 2005.

D. Erdogmus, K. E. Hild, and J. C. Principe. Independent Component Analysis using Renyi's Mutual Information and Legendre Density Estimation. In *Proceedings of International Joint Conference on Neural Networks*, pages 2762–2767, Washington, USA, July 15-19, 2001.

D. Erdogmus, K. E. Hild, and J. C. Principe. Blind Source Separation using Renyi's α-Marginal Entropies. *Neurocomputing*, 49:25–38, 2002.

D. Erdogmus, K. E. Hild, J. C. Principe, M. Lazaro, and I. Santamaria. Adaptive Blind Deconvolution of Linear Channels using Renyi's Entropy with Parzen Window Estimation. *IEEE Transactions on Signal Processing*, 52(6):1489–1498, 2004a.

D. Erdogmus, K. E. Hild, Y. N. Rao, and J. C. Principe. Minimax Mutual Information Approach for Independent Component Analysis. *Neural Computation*, 16:1235–1252, 2004b.

D. Erdogmus and J. C. Principe. An Error-Entropy Minimization Algorithm for Supervised Training of Nonlinear Adaptive Systems. *IEEE Transactions on Signal Processing*, 50(7): 1780–1786, 2002a.

D. Erdogmus and J. C. Principe. Generalized Information Potential Criterion for Adaptive System Training. *IEEE Transactions on Neural Networks*, 13(5):1035–1044, 2002b.

D. Erdogmus and J. C. Principe. Convergence Properties and Data Efficiency of the Minimum Error-Entropy Criterion in Adaline Training. *IEEE Transactions on Signal Processing*, 51(7):1966–1978, 2003.

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, USA, August 2-4, 1996.

R. Faltlhauser and G. Ruske. Robust Speaker Clustering in Eigenspace. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 57–60, Madonna di Campiglio Trento, Italy, December 9-13, 2001.

R. Fano. *Transmission of Information*. MIT Press, Cambridge, 1961.

M. Fiedler. Algebraic Connectivity in Graphs. *Czechoslovak Mathematics Journal*, 23:298–305, 1973.

D. J. Field. What is the Goal of Sensory Coding? *Neural Computation*, 6:559–601, 1994.

A. L. N. Fred and A. K. Jain. Robust Data Clustering. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 128–136, Madison, USA, June 16-22, 2003.

N. Friedman, O. Mosenzon, N. Slonim, and N. Tishby. Multivariate Information Bottleneck. In *Proceedings of Conference on Uncertainty in Artificial Intelligence*, pages 151–161, Seattle, USA, August 2-5, 2001.

K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 1990.

M. Gaeta and J.-L. Lacoume. Source Separation without Prior Knowledge: The Maximum Likelihood Solution. In *Proceedings of the European Signal Processing Conference*, pages 621–624, Barcelona, Spain, September 18-21, 1990.

J. Goddard, A. E. Martinez, F. M. Martinez, and T. Aljama. A Comparison of Different Clustering Algorithms for Speech Recognition. In *Proceedings of IEEE Midwest Symposium on Circuits and Systems*, volume 3, pages 1222–1225, Lansing, USA, August 8-11, 2000.

E. Gokcay and J. Principe. A New Clustering Evaluation Function using Renyi's Information Potential. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pages 3490–3493, Istanbul, Turkey, June 6-9, 2000.

E. Gokcay and J. Principe. Information Theoretic Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):158–170, 2002.

D. Gondek and T. Hofmann. Conditional Information Bottleneck Clustering. In *Workshop on Clustering Large Data Sets, IEEE International Conference on Data Mining*, Melbourne, USA, November 19-22, 2003.

D. Greene, A. Tsymbal, N. Bolshakova, and P. Cunningham. Ensemble Clustering in Medical Diagnostics. In *Proceedings of IEEE Symposium on Computer-Based Medical Systems*, pages 576–581, Bethesda, USA, June 24-25, 2004.

G. D. Guo and S. D. Ma. Unsupervised Segmentaton of Color Images. In *Proceedings of International Conference on Image Processing*, pages 299–302, Chicago, USA, October 4-7, 1998a.

G. D. Guo and S. D. Ma. Unsupervised Segmentaton of SAR Images. In *Proceedings of International Geoscience and Remote Sensing Symposium*, volume 2, pages 1150–1152, Seattle, USA, July 6-10, 1998b.

L. Hagen and A. B. Kahng. Fast Spectral Methods for Ratio Cut Partitioning and Clustering. In *Proceedings of IEEE International Conference on Computer-Aided Design*, pages 10–13, Santa Clara, USA, November 11-14, 1991.

J. A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, New York, 1975.

R. V. Hartley. Transmission of Information. *Bell System Technical Journal*, 7:535–563, 1928.

T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The Entire Regularization Path for the Support Vector Machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

R. J. Hathaway and J. C. Bezdek. Local Convergence of the Fuzzy C-Means Algorithm. *Pattern Recognition*, 19(6):477–480, 1986.

R. J. Hathaway, J. W. Davenport, and J. C. Bezdek. Relational Duals of the C-Means Clustering Algorithms. *Pattern Recognition*, 22(2):205–212, 1989.

S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, New Jersey, 2nd edition, 1999.

S. Haykin, editor. *Unsupervised Adaptive Filtering: Volume 1, Blind Source Separation*. John Wiley & Sons, New York, 2000.

S. Haykin. *Adaptive Filter Theory*. Prentice Hall, New Jersey, 4th edition, 2002.

X. He, H. Zha, C. Ding, and H. D. Simon. Web Document Clustering using Hyperlink Structures. *Computational Statistics and Data Analysis*, 45:19–45, 2002.

Z. He, L. Yang, J. Liu, Z. Lu, C. He, and Y. Shi. Blind Source Separation using Clustering based Multivariate Density Estimation Algorithm. *IEEE Transactions on Signal Processing*, 48(2):575–579, 2000.

J. Herrero, A. Valencia, and J. A. Dopazo. A Hierarchical Unsupervised Growing Neural Network for Clustering Gene Expression Patterns. *Bioinformatics*, 17:126–136, 2001.

D. J. Higham and M. Kibble. A Unified View of Spectral Clustering. Technical Report 2, University of Strathclyde, Department of Mathematics, January 2004.

K. E. Hild. *Blind Separation of Convolutive Mixtures using Renyi's Divergence*. PhD thesis, University of Florida, Gainesville, FL, USA, 2003.

K. E. Hild, D. Erdogmus, and J. C. Principe. Blind Source Separation using Renyi's Mutual Information. *IEEE Signal Processing Letters*, 8(6):174–176, 2001.

K. E. Hild, D. Erdogmus, and J. C. Principe. Sequential Feature Extraction using Information-Theoretic Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence, submitted*, 2005a.

K. E. Hild, D. Pinto, D. Erdogmus, and J. C. Principe. Blind Source Separation using Information-Theoretic Learning. *Signal Processing, submitted*, 2005b.

K. E. Hild, D. Pinto, D. Erdogmus, and J. C. Principe. Convolutive Blind Source Separation by Minimizing Mutual Information between Segments of Processes. *IEEE Transactions on Circuits and Systems I: Theory and Applications, submitted*, 2005c.

A. Hinneburg and D. A. Keim. An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pages 58–65, New York, USA, August 27-31, 1998.

T. Hofmann and J. M. Buhmann. Pairwise Data Clustering by Deterministic Annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):1–14, 1997.

A. Honkela and H. Valpola. Variational Learning and Bits-Back Coding: An Information-Theoretic View to Bayesian Learning. *IEEE Transactions on Neural Networks*, 15(4):800–810, 2004.

P. Hoyer and A. Hyvärinen. Independent Component Analysis Applied to Feature Extraction from Colour and Stereo Images. *Network: Computation in Neural Systems*, 11(3):191–210, 2000.

S. Huang, G.-R. Xue, B.-Y. Zhang, Z. Chen, and Y. Yu. Multi-Type Features based Web Document Clustering. In *Proceedings of International Conference on Web Information Systems Engineering*, pages 253–265, Brisbane, Australia, November 22-24, 2004.

M. M. Van Hulle. Entropy-Based Mixture Modeling for Topographic Map Formation. *IEEE Transactions on Neural Networks*, 15(4):850–858, 2004.

J. Hurri. Independent Component Analysis of Image Data. Master's thesis, Helsinki University of Technology, 1997.

J. Hurri, A. Hyvärinen, and E. Oja. Wavelets and Natural Image Statistics. In *Proceedings of Scandinavian Conference on Image Analysis*, Lappeenranta, Finland, June 9-11, 1997.

A. Hyvärinen. Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999a.

A. Hyvärinen. Sparse Code Shrinkage: Denoising of Non-Gaussian Data by Maximum Likelihood Estimation. *Neural Computation*, 11(7):1739–1768, 1999b.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, New York, 2001.

A. Hyvärinen and E. Oja. A Fast Fixed-Point Algorithm for Independent Component Analysis. *Neural Computation*, 9:1483–1492, 1997.

A. Hyvärinen and E. Oja. Independent Component Analysis by General Nonlinear Hebbian-like Learning Rules. *Signal Processing*, 64(3):310–313, 1998.

N. Intrator. Feature Extraction using an Unsupervised Neural Network. *Neural Computation*, 4:98–107, 1992.

M. A. Ismail and S. Z. Selim. Fuzzy C-Means: Optimality of Solutions and Effective Termination of the Algorithm. *Pattern Recognition*, 19(6):481–485, 1986.

K. Iwata, K. Ikeda, and H. Sakai. A New Criterion Using Information Gain for Action Selection Strategy in Reinforcement Learning. *IEEE Transactions on Neural Networks*, 15(4): 792–799, 2004.

A. J. Izenman. Recent Developments in Nonparametric Density Estimation. *Journal of the American Statistical Association*, 86(413):205–224, 1991.

A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Englewood Cliffs, 1988.

A. K. Jain, R. P. W. Duin, and J. Mao. Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, 31(3):264–323, 1999.

E. T. Jaynes. Information Theory and Statistical Mechanics. *The Physical Review*, 106(4):620–630, 1957.

H. Jeffreys. *Theory of Probability*. Oxford University Press, Oxford, 1948.

R. Jenssen and T. Eltoft. ICA Filter Bank for Segmentation of Textured Images. In *Proceedings of International Workshop on Independent Component Analysis and Blind Signal Separation*, pages 827–832, Nara, Japan, April 1-4, 2003.

R. Jenssen, T. Eltoft, and J. C. Principe. Information Theoretic Clustering: A Unifying Review of Three Recent Algorithms. In *Proceedings of Nordic Signal Processing Symposium*, pages 292–295, Espoo, Finland, June 9-11, 2004a.

R. Jenssen, T. Eltoft, and J. C. Principe. Information Theoretic Spectral Clustering. In *Proceedings of International Joint Conference on Neural Networks*, pages 111–116, Budapest, Hungary, July 25-29, 2004b.

R. Jenssen, D. Erdogmus, K. E. Hild, J. C. Principe, and T. Eltoft. Information Force Clustering using Directed Trees. In *Proceedings of International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 68–72, Lisbon, Portugal, July 7-9, 2003a.

R. Jenssen, D. Erdogmus, J. C. Principe, and T. Eltoft. Towards a Unification of Information Theoretic Learning and Kernel Methods. In *Proceedings of IEEE Workshop on Machine Learning for Signal Processing*, pages 1–8, Sao Luis, Brazil, September 29 - October 1, 2004c.

R. Jenssen, D. Erdogmus, J. C. Principe, and T. Eltoft. The Laplacian PDF Distance: A Cost Function for Clustering in a Kernel Feature Space. In *Advances in Neural Information Processing Systems 17 (to appear)*, MIT Press, Cambridge, 2005.

R. Jenssen, K. E. Hild, D. Erdogmus, J. C. Principe, and T. Eltoft. Clustering using Renyi's Entropy. In *Proceeding of International Joint Conference on Neural Networks*, pages 523–528, Portland, USA, July 20-24, 2003b.

R. Jenssen, T. A. Øigård, T. Eltoft, and A. Hanssen. Sparse Code Shrinkage Based on the Normal Inverse Gaussian Density Model. In *Proceedings of International Workshop on Independent Component Analysis and Blind Signal Separation*, pages 212–217, San Diego, USA, December 9-13, 2001.

R. Jenssen, J. C. Principe, and T. Eltoft. Cauchy-Schwartz pdf Divergence Measure for non-Parametric Clustering. In *Proceedings of IEEE Norway Section Signal Processing Symposium*, Bergen, Norway, October 2-4, 2003c.

R. Jenssen, J. C. Principe, and T. Eltoft. Information Cut and Information Forces for Clustering. In *Proceedings of IEEE International Workshop on Neural Networks for Signal Processing*, pages 459–468, Toulouse, France, September 17-19, 2003d.

M. C. Jones, J.S. Marron, and S. J. Sheater. A Brief Survey of Bandwidth Selection for Density Estimation. *Journal of the Royal Statistical Society*, 87:227–233, 1996.

T.P. Jung, C. Humphries, T.-W. Lee, S. Makeig, M-J. McKeown, V. Iragui, and T. Sejnowski. Extended ICA Removes Artifacts from Electroencephalographic Recordings. In *Advances in Neural Information Processing Systems, 10*, MIT Press, Cambridge, 1998.

C. Jutten. Source Separation: From Dusk Till Dawn. In *Proceedings of International Workshop on Independent Component Analysis and Blind Source Separation*, pages 15–26, Helsinki, Finland, June 3-6, 2000.

R. Kannan, S. Vempala, and A. Vetta. On Clusterings: Good, Bad and Spectral. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, pages 367–377, Redondo Beach, USA, November 12-14, 2000.

J. N. Kapur. *Maximum Entropy Models in Science and Engineering*. John Wiley & Sons, New York, 1989.

J. N. Kapur. *Measures of Information and Their Applications*. John Wiley & Sons, New York, 1994.

J. N. Kapur and H. K. Kesavan. *Entropy Optimization Principles with Applications Models in Science and Engineering*. Academic Press, Boston, 1992.

D. Kazakos and P. Papantoni-Kazakos. *Detection and Estimation*. Computer Science Press, New York, 1990.

B. King. Step-Wise Clustering Procedures. *Journal of the American Statistical Association*, pages 86–101, 1967.

J. Kogan, C. Nicholas, and V. Volkovich. Text Mining with Information Theoretic Clustering. *Computing in Science and Engineering*, 5(6):52–59, 2003.

T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, 1989.

A. Kraskov, H. Stogbauer, R. G. Andrzejak, and P. Grassberger. Hierarchical Clustering based on Mutual Information. *Bioinformatics (to appear)*, 2004.

S. Kulkarni, G. Lugosi, and S. Venkatesh. Learning Pattern Classification - A Survey. In *Information Theory - 50 Years of Discovery*, pages 2178–2206, S. Verdu and S. McLaughlin (Eds.), IEEE Press, Piscataway, 2000.

S. Kullback. *Information Theory and Statistics*. Dover Publications, New York, 1968.

S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

M. Lazaro, I. Santamaria, D. Erdogmus, K. E. Hild II, C. Pantaleon, and J. C. Principe. Stochastic Blind Equalization Based on PDF Fitting using Parzen Estimator. *IEEE Transactions on Signal Processing*, 53(2):696–704, 2005.

E. Learned-Miller and J. W. Fisher. ICA using Spacing Estimates of Entropy. *Journal of Machine Learning Research*, 4:1271–1295, 2003.

Y. A. LeCun, L. D. Jackel, L. Bottou, A. Brunot, C. Cortes, J.S. Denker, H. Drucker, I. Guyon, U. A. Müller, E. Säckinger, P. Y. Simard, and V. N. Vapnik. Learning Algorithms for Classification: A Comparison on Handwritten Digit Reconstruction. *Neural Networks*, pages 261–276, 1995.

T.-W. Lee, T. P. Jung, S. Makeig, and T. Sejnowski, editors. *Proceedings of the Third International Conference on Independent Component Analysis and Blind Signal Separation*, San Diego, USA, December 9-12, 2001.

H. Li, K. Zhang, and T. Jiang. Minimum Entropy Clustering and Applications to Gene Expression Analysis. In *Proceedings of IEEE Computational Systems Bioinformatics Conference*, pages 142–151, Stanford, USA, August 16-19, 2004.

Y. Linde, A. Buzo, and R. M. Gray. An Algorithm for Vectors Quantizer Design. *IEEE Transactions on Communications*, 28:84–95, 1980.

S. P. Lloyd. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28 (2):129–137, 1982.

P. Lopez-Martinez, L. van Kempen, H. Sahli, and D. Cabello Ferrer. Improved Thermal Analysis of Buried Landmines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(9): 1965–1975, 2004.

J. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, University of California Press, Berkeley, 1967.

S. Malaroiu, K. Kiviluoto, and E. Oja. Time Series Prediction with Independent Component Analysis. In *Proceedings of International Conference on Advanced Investment Technology*, Gold Cost, Australia, December 19-21, 1999.

J. Mao and A. K. Jain. A Self-Organizing Network for Hyperellipsoidal Clustering (HEC). *IEEE Transactions on Neural Networks*, 7:16–29, 1996.

G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

G. J. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, New York, 2000.

M. Meila and L. Xu. Multiway Cuts and Spectral Clustering. Technical Report 442, University of Washington, Department of Statistics, January 2004.

J. Mielikainen and P. Toivanen. Clustered DPCM for the Lossless Compression of Hyperspectral Images. *IEEE Transactions on Geoscience and Remote Sensing*, 41(12):2943–2946, 2003.

S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller. Fisher Discriminant Analysis with Kernels. In *Proceedings of IEEE International Workshop on Neural Networks for Signal Processing*, pages 41–48, Madison, USA, August 23-25, 1999.

P. P. Mohanta, D. P Mukherjee, and S. T. Acton. Agglomerative Clustering for Image Segmentation. In *Proceedings of International Conference on Pattern Recognition*, volume 1, pages 664–667, Quebec, Canada, August 11-15, 2002.

R. A. Morejon and J. C. Principe. Advanced Search Algorithms for Information Theoretic Learning with Kernel-Based Estimators. *IEEE Transactions on Neural Networks*, 15(4):874–884, 2004.

K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An Introduction to Kernel-Based Learning Algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

K. R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. N. Vapnik. Predicting Time Series with Support Vector Machines. In *Proceedings of International Conference on Artificial Neural Networks - Lecture Notes in Computer Science, Springer-Verlag*, volume 1327, pages 999–1004, Berlin, 1997.

F. Murtagh. A Survey of Recent Advances in Hierarchical Clustering Algorithms which use Cluster Centers. *Computer Journal*, 26(4):354–359, 1984.

A. Y. Ng, M. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems, 14*, pages 849–856, MIT Press, Cambridge, 2002.

A. Y. Ng, A. X. Zheng, and M. Jordan. Link Analysis, Eigenvectors and Scalability. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 903–910, Seattle, USA, August 4-10, 2001.

D. Obradovic and G. Deco. Information Maximization and Independent Component Analysis: Is There a Difference? *Neural Computation*, 10(8), 1998.

B. A. Olshausen and D. J. Field. Natural Image Statistics and Efficient Coding. *Network: Computation in Neural Systems*, 7(2), 1996.

P. Pajunen and J. Karhunen, editors. *Proceedings of the Second International Workshop on Independent Component Analysis and Blind Signal Separation*, Helsinki, Finland, June 19-22, 2000.

E. Parzen. On the Estimation of a Probability Density Function and the Mode. *The Annals of Mathematical Statistics*, 32:1065–1076, 1962.

K. Pawelzik, K.-R. Müller, and J. Kohlmorgen. Prediction of Mixtures. In *Proceedings of International Conference on Artificial Neural Networks - Lecture Notes in Computer Science*, volume 1112, pages 127–132, Berlin, 1996. Springer-Verlag.

F. Perez-Cruz and O. Bousquet. Kernel Methods and Their Potential Use in Signal Processing. *IEEE Signal Processing Magazine*, pages 57–65, May 2004.

P. Perona and W. T. Freeman. A Factorization Approach to Grouping. In *Proceedings of European Conference on Computer Vision*, pages 655–670, Freiburg im Breisgay, Germany, June 2-6, 1998.

D. T. Pham. Blind Separation of Instantaneous Mixture Sources via an Independent Component Analysis. *Network: Computation in Neural Systems*, 7(2), 1996.

D. T. Pham, P. Garrat, and C. Jutten. Separation of a Mixture of Independent Sources through a Maximum Likelihood Approach. In *Proceedings of the European Signal Processing Conference*, pages 771–774, Brussels, Belgium, August 24-27, 1992.

A. Pothen, H. D. Simon, and K. P. Liou. Partitioning Sparse Matrices with Eigenvectors of Graphs. *SIAM Journal of Matrix Analysis and Applications*, 11(3):430–452, 1990.

J. Principe, D. Xu, and J. Fisher. Information Theoretic Learning. In *Unsupervised Adaptive Filtering*, volume I, S. Haykin (Ed.), John Wiley & Sons, New York, 2000a. Chapter 7.

J. C. Principe, E. Oja, L. Xu, A. Cichocki, and D. Erdogmus. Guest Editorial - Special Issue on Information Theoretic Learning. *IEEE Transactions on Neural Networks*, 15(4):789–791, 2004.

J. C. Principe, D. Xu, Q. Zhao, and J. W. Fisher. Learning From Examples with Information Theoretic Criteria. *Journal of VLSI Signal Processing*, 26(1):61–77, 2000b.

A. Renyi. On Measures of Entropy and Information. *Selected Papers of Alfred Renyi, Akademiai Kiado, Budapest*, 2:565–580, 1976a.

A. Renyi. Some Fundamental Questions of Information Theory. *Selected Papers of Alfred Renyi, Akademiai Kiado, Budapest*, 2:526–552, 1976b.

J. Rigau, M. Feixas, A. Bardera, and I. Boada. Medical Image Segmentation based on Mutual Information Maximization. In *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 135–142, Saint-Malo, France, September 26-29, 2004.

S. J. Roberts, R. Everson, and I. Rezek. Minimum Entropy Data Partitioning. In *Proceedings of IEE International Conference on Artificial Neural Networks*, volume 2, pages 844–849, London, UK, September 7-10, 1999.

S. J. Roberts, R. Everson, and I. Rezek. Maximum Certainty Data Partitioning. *Pattern Recognition*, 33:833–839, 2000.

S. J. Roberts, C. Holmes, and D. Denison. Minimum Entropy Data Partitioning using Reversible Jump Markov Chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):909–914, 2001.

K. Rose, E. Gurewitz, and G. C. Fox. A Deterministic Annealing Approach to Clustering. *Pattern Recognition Letters*, 11(11):589–594, 1990.

K. Rose, E. Gurewitz, and G. C. Fox. Vector Quantization by Deterministic Annealing. *IEEE Transactions on Information Theory*, 38(4):1249–1257, 1992.

K. Rose, E. Gurewitz, and G. C. Fox. Constrained Clustering as an Oprimization Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(8):785–794, 1993.

K. Rose, D. Miller, and A. Gersho. Entropy-Constrained Tree-Structured Vector Quantizer Design by the Minimum Cross Entropy Principle. In *Proceedings of IEEE Data Compression Conference*, pages 12–21, Snowbird, Utah, March 29-31, 1994.

V. Roth and V. Steinhage. Nonlinear Discriminant Aanalysis using Kernel Functions. In *Advances in Neural Information Processing Systems 12,*, pages 568–574, MIT Press, Cambridge, 2000.

S. Roweis and L. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290:2323–2326, 2000.

L. Rutkowski. Adaptive Probabilistic Neural Networks for Pattern Classification in Time-Varying Environment. *IEEE Transactions on Neural Networks*, 15(4):811–827, 2004.

M. A. Sanchez-Montanes and F. J. Corbacho. A New Information Processing Measure for Adaptive Complex Systems. *IEEE Transactions on Neural Networks*, 15(4):917–927, 2004.

I. Santamaria, D. Erdogmus, and J. C. Principe. Entropy Minimization for Supervised Digital Communications Channel Equalization. *IEEE Transactions on Signal Processing*, 50(5):1184–1192, 2002.

S. Sarkar and P. Soundararajan. Supervised Learning of Large Perceptual Organization: Graph Spectral Partitioning and Learning Automata. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):504–525, 2000.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, 2002.

B. Schölkopf, A. J. Smola, and K. R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10:1299–1319, 1998.

N. N. Schraudolph. Gradient-Based Manipulation of Nonparametric Entropy Estimates. *IEEE Transactions on Neural Networks*, 15(4):828–837, 2004.

D. W. Scott. *Multivariate Density Estimation*. John Wiley & Sons, New York, 1992.

G. Scott and H. Longuet-Higgins. Feature Grouping by Relocalisation of Eigenvectors of the Proximity Matrix. In *Proceedings of British Machine Vision Conference*, pages 103–108, Oxford, UK, September 24-27, 1990.

S. Z. Selim and M. A. Ismail. K-Means Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1):81–87, 1984.

C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27: 379–423, 623–653, 1948.

C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, 1949.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

S. J. Sheater and M. C. Jones. A Reliable Data-Based Bandwidth Selection Method for Kernel Density Estimation. *Journal of the Royal Statistical Society, Ser. B*, 53:683–690, 1991.

S. Shen, W. A. Sandham, and M. H. Granat. Preprocessing and Segmentation of Brain Magnetic Resonance Images. In *Proceedings of IEEE Conference on Information Technology Applications in Biomedicine*, pages 149–152, Birmingham, UK, April 24-26, 2003.

J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

S. Shimoji and S. Lee. Data Clustering with Entropical Scheduling. In *Proceedings of IEEE International Conference on Neural Networks*, volume 4, pages 2423–2428, Orlando, USA, June 26 - July 2, 1994.

K. Siala and A. Benazza-Benyahia. Hyperspectral Image Compression through Spectral Clustering. In *Proceedings of International Symposium on Control, Communications and Signal Processing*, volume 1, pages 435–438, Hammamet, Tunisia, March 21-24, 2004.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, 1986.

V. Sindhwani, S. Rakshit, D. Deodhar, D. Erdogmus, J. C. Principe, and P. Niyogi. Feature Selection in MLPs and SVMs based on Maximum Ooutput Information. *IEEE Transactions on Neural Networks*, 15(4):937–948, 2004.

N. Slonim and N. Tishby. Agglomerative Information Bottleneck. In *Advances in Neural Information Processing Systems, 12*, pages 617–623, MIT Press, Cambridge, 2000.

P. H. A. Sneath and R. R. Sokal. *Numerical Taxonomy*. Freeman, London, 1973.

E. Sorouchyari. Blind Separation of Sources, Part III: Stability Analysis. *Signal Processing*, 24: 21–29, 1991.

S. Still, W. Bialek, and L. Bottou. Geometric Clustering using the Information Bottleneck Method. In *Advances in Neural Information Processing Systems, 16 (to appear)*, MIT Press, Cambridge, 2004.

M. J. Symon. Clustering Criterion and Multi-Variate Normal Mixture. *Biometrics*, 77:35–43, 1977.

J. Tenenbaum, V. de Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290:2319–2323, 2000.

S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, San Diego, 1999.

N. Tishby, F. C. Pereira, and W. Bialek. The Information Bottleneck Method. In *Proceedings of Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, Monticello, USA, September 22-24, 1999.

N. Tishby and N. Slonim. Data Clustering by Markovian Relaxation and the Information Bottleneck Method. In *Advances in Neural Information Processing Systems, 13*, pages 640–646, MIT Press, Cambridge, 2001.

K. Torkkola. Blind Separation for Audio Signals - Are We There Yet? In *Proceedings of International Workshop on Independent Component Analysis and Blind Signal Separation*, pages 239–244, Aussois, France, January 11-15, 1999.

J. H. van Hateren and A. van der Schaaf. Independent Component Filters of Natural Images Compared with Simple Cells in Primary Visual Cortex. *Proceedings of the Royal Society Ser. B*, 256:359–366, 1998.

V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

S. Verdu. Fifty Years of Shannon Theory. In *Information Theory - 50 Years of Discovery*, pages 2178–2206, S. Verdu and S. McLaughlin (Eds.), IEEE Press, Piscataway, 2000.

D. Verma and M. Meila. A comparison of Spectral Clustering Algorithms. Technical Report 03-05-01, University of Washington, Department of Statistics, 2003.

R. Vigário. Extraction of Ocular Artifacts from EEG using Independent Component Analysis. *Electroenceph. Clin. Neurophysiol.*, 103(3):395–404, 1997.

R. Vigário, V. Jousmäki, M. Hämäläinen, R. Hari, and E. Oja. Independent Component Analysis for Identification of Artifacts in Magnetoencephalographic Recordings. In *Advances in Neural Information Processing Systems, 10*, pages 229–235, MIT Press, Cambridge, 1998.

R. Vigário, J. Särelä, V. Jousmäki, M. Hämäläinen, and E. Oja. Independent Component Approach to the Analysis of EEG and MEG Recordings. *IEEE Transactions on Biomedical Engineering*, 47(5):589–593, 2000.

P. Viola and W. M. Wells. Alignment by Maximization of Mutual Information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

P. A. Viola, N. N. Schraudolph, and T. J. Sejnowski. Empirical Entropy Manipulation for Real-World Problems. In *Advances in Neural Information Processing Systems, 8*, pages 851–857, MIT Press, Cambridge, 1995.

M. P. Wand and M. C. Jones. *Kernel Smooting*. Chapman and Hall, London, 1995.

S. Wang, D. Schuurmans, F. Peng, and Y. Zhao. Learning Mixture Models with the Regularized Latent Maximum Entropy Principle. *IEEE Transactions on Neural Networks*, 15(4): 903–916, 2004.

J. H. Ward. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58:236–244, 1963.

S. Watanabe, Y. Minami, A. Nakamura, and N. Ueda. Variational Bayesian Estimation and Clustering for Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, 12(4): 365–381, 2004.

Y. Weiss. Segmentation Using Eigenvectors: A Unifying View. In *Proceedings of IEEE International Conference on Computer Vision*, pages 975–982, Corfu, Greece, September 20-25, 1999.

B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Prentice Hall, New Jersey, 1985.

D. Xu. *Energy, Entropy and Information Potential for Neural Computation*. PhD thesis, University of Florida, Gainesville, FL, USA, 1999.

D. Xu, J. Fisher, J. C. Principe, and H. C. Wu. A Novel Measure for Independent Component Analysis. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pages 1161–1164, Seattle, USA, May 12-15, 1998.

C. T. Zahn. Graph Theoretic Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers*, 20:68–86, 1971.

H.-J. Zeng, Z. Chen, and W.-Y. Ma. A Unified Framework for Clustering Heterogeneous Web Objects. In *Proceedings of International Conference on Web Information Systems Engineering*, pages 161–172, Singapore, December 12-14, 2002.

H. Zha, X. He, C. Ding, H. Simon, and M. Gu. Spectral Relaxation for K-means Clustering. In *Advances in Neural Information Processing Systems, 14*, pages 1057–1064, MIT Press, Cambridge, 2002.

X. Zhou, X. Wang, E. R. Dougherty, D. Russ, and E. Suh. Gene Clustering Based on Cluster-wide Mutual Information. *Journal of Computational Biology*, 1:147–161, 2004.

A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K. R. Müller. Engineering Support Vector Machine Kernels that Recognize Translation Invariant Sites in DNA. *Bioinformatics*, 16:906–914, 2000.

About the author - ROBERT JENSSEN comes from a place called Ersfjorden near the city of Tromsø, Norway. In December 2000, he received the M.Sc. degree in electrical engineering from the University of Tromsø, which is the northernmost unversity in the world. In may 2005, he defended the current Ph.D. thesis, entitled "An Information Theoretic Approach to Machine Learning." The contents of this thesis reflects Jenssen's research interests, which are in the areas of information theoretic learning, kernel methods, spectral clustering and independent component analysis.

Jenssen received Honorable Mention for the year 2003 Pattern Recognition Journal Best Paper Award, for the paper "Independent Component Analysis for Texture Segmentation," co-authored with Torbjørn Eltoft. Jenssen also received the ICASSP 2005 Best Student Paper Award, for the paper "The Laplacian Spectral Classifier."