Information Theoretic Angle-Based Spectral Clustering: A Theoretical Analysis and an Algorithm

Robert Jenssen*, *Member, IEEE*, Deniz Erdogmus, *Member, IEEE*, Jose C. Principe *Fellow, IEEE* and Torbjørn Eltoft, *Member, IEEE*

***Abstract*— Recent work has revealed a close connection between certain information theoretic divergence measures and properties of Mercer kernel feature spaces. Specifically, it has been proposed that an information theoretic measure may be used as a cost function for clustering in a kernel space, approximated by the spectral properties of the Laplacian matrix. In this paper we extend this result to other kernel matrices. We develop an algorithm for the actual clustering which is based on comparing angles between data points, and demonstrate that the proposed method performs equally good as a state-of-the art spectral clustering method. We point out some drawbacks of spectral clustering related to outliers, and suggest measures to be taken.**

I. Introduction

Recently, close connections between information theoretic learning [1], [2], [3] and Mercer kernel methods [4], [5], [6] have been revealed [7], [8]. Information theoretic learning refers in this regard to adaptive systems training using performance criteria related to Renyi's measure of entropy [9], such as for example a divergence measure based on the Cauchy-Schwarz inequality. These measures are combined with non-parametric density estimation using Parzen windowing [10]. Mercer kernel methods are based on a non-linear data mapping to a feature space, where inner-products can be implicitly computed based on the input data using a kernel function.

In [8] it was shown that a Cauchy-Schwarz based divergence measure between probability density functions (pdfs) has a dual expression as a measure of the cosine of the angle between cluster mean vectors in a Mercer kernel feature space. It was pointed out that the data mapping to the kernel space might be approximated by the eigenspectrum, i.e. the eigenvalues and eigenvectors, of the Laplacian data matrix. Thus, potentially enabling an angle-based spectral clustering procedure to be developed, using an information theoretic cost function. However, no such algorithm was implemented.

Spectral clustering dates back at least to [11], who discovered that a graph can be bi-partitioned by thresholding the eigenvector corresponding to the second eigenvalue of the Laplacian matrix. Recently, a number of related techniques have been proposed [12], [13], [14], [15], [16], [17]. It is however not always clear which criterion that is optimized by spectral clustering in terms of the input data set, although most of these techniques can be shown to be approximate solutions to the minimum graph cut problem. In Ng et al. [18] the eigenvectors of the Laplacian matrix are used to transform, or map, the input data into a new representation. Then, the actual clustering in that space is performed using the C-means technique [19]. Provided the neighborhood parameter, or kernel size, used to construct the Laplacian matrix is appropriate, the Ng et al. method has been shown exhibit excellent performance.

The contribution made in this paper is threefold. First, we extend and generalize the theoretical results obtained in [8]. We consider a Cauchy-Schwarz based divergence measure between pdfs. The building blocks of the divergence measure are inner-products, which are defined in terms of a weighting function $u(\mathbf{x})$. We show that the divergence measure has a dual expression as a measure of the cosine of the angle between cluster mean vectors in a Mercer kernel feature space defined by $u(\mathbf{x})$. The data mapping to this kernel space may be approximated by the eigenspectrum of a kernel matrix $\mathbf{K}_u$. For a particular $u(\mathbf{x})$, the special case considered in [8] is obtained. Second, we derive a clustering algorithm in the $u(\mathbf{x})$-dependent Mercer kernel feature space. In terms of the input data set, this algorithm is founded on the well defined information theoretic optimality criterion. Since this measure corresponds to an angle measure in the kernel space, the structure of the algorithm in that space is very simple, consisting of comparing angles between feature space data points and feature space mean vectors. We show that the information theoretic angle-based spectral clustering algorithm performs equally good as the state-of-the art Ng et al. spectral clustering method. Third, the introduction of the weighting function $u(\mathbf{x})$ allows us to identify a situation where spectral clustering may break down. This situation may occur when there are outliers in the data set. We indicate that the weighting function may be actively used to avoid this problem.

In section II, we outline the theory connecting the information theoretic cost function to a Mercer kernel feature space. The actual clustering algorithm is derived in section III. Thereafter, we present some clustering experiments in section IV. In section V, we discuss the problem of outliers an illustrate with examples. Finally, we conclude the paper in section VI.

*Corresponding author. Email: robertj@phys.uit.no, Tel.: (47) 77646493, Fax: (47) 77645580

Robert Jenssen and Torbjørn Eltoft are with the Department of Physics, University of Tromsø, Tromsø, N-9037, Norway (email Eltoft: pcte@phys.uit.no).

Deniz Erdogmus is with the Department of Computer Science and Engineering, Oregon Graduate Institute, OHSU, Portland, OR 97006, USA (email: derdogmus@ieee.org).

Jose C. Principe is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA (email: principe@cnel.ufl.edu).

II. An Information Theoretic Cost Function in a Mercer Kernel Space

This section provides an extension of the theory presented in [8]. We show that an information theoretic divergence measure between pdfs has a dual expression as a divergence measure between clusters in a Mercer kernel feature space, whose properties depends on the choice of an inner-product weighting function.

We consider a Cauchy-Schwarz based divergence measure between the pdfs $p_1(\mathbf{x}), \ldots, p_C(\mathbf{x})$. We define it as

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

where $\kappa = \sum_{c=1}^{C}$ is a normalizing constant. The weighted inner-products are given by $\langle p_i, p_j \rangle_u \equiv \int p_i(\mathbf{x}) p_j(\mathbf{x}) u(\mathbf{x}) d\mathbf{x}$, $i, j = 1, \ldots, C$, where $u(\mathbf{x})$ is some non-negative weighting function. If $p_1(\mathbf{x}) = \cdots = p_C(\mathbf{x})$, then $D(p_1, \ldots, p_C) = 0$.

Note that for $C = 2$ and $u(\mathbf{x}) \equiv 1$, this divergence measure reduces to the measure used by Principe [1] for adaptive systems training, e.g. for independent component analysis. Note also that the case $u(\mathbf{x}) = f^{-1}(\mathbf{x})$, where $f(\mathbf{x})$ is the overall pdf of the data set, corresponds to the Laplacian measure analyzed in [8]. In the following, we will analyze this divergence measure for a general $u(\mathbf{x})$, using Parzen windowing [10] to estimate probability densities.

Parzen windowing is a well-known kernel-based density estimation method [20], [10]. Given a set of d-dimensional 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

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

Here, W_σ is the Parzen window, or kernel, and σ 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,

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

It is easily shown that Eq. (2) is an asymptotically unbiased and consistent estimator provided σ decays to zero at a certain rate as N tends to infinity [10]. In the finite sample case, the kernel size has to be chosen in a trade-off between estimation bias and variance. In order to obtain analytical results, we will consider Gaussian Parzen windows in the following. This is however not necessary for the validity of the results, other window functions may be used.

We will analyze the Cauchy-Schwarz measure for the case $C = 2$. The extension to more than two pdfs is straightforward. Then, we have

$$D(p_1, p_2) = -\log \frac{\int h(\mathbf{x}) g(\mathbf{x}) d\mathbf{x}}{\sqrt{\int h^2(\mathbf{x}) d\mathbf{x} \int g^2(\mathbf{x}) d\mathbf{x}}}, \tag{3}$$

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 pdf, or 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 overall 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. The following generalized Parzen window-based estimator for the functions $h(\mathbf{x})$ and $g(\mathbf{x})$ may be used

$$\hat{h}(\mathbf{x}) = \frac{1}{N_1} \sum_{i=1}^{N_1} u^{\frac{1}{2}}(\mathbf{x}_i) W_\sigma(\mathbf{x}, \mathbf{x}_i), \tag{4}$$

$$\hat{g}(\mathbf{x}) = \frac{1}{N_2} \sum_{j=1}^{N_2} u^{\frac{1}{2}}(\mathbf{x}_j) W_\sigma(\mathbf{x}, \mathbf{x}_j), \tag{5}$$

where we for simplicity have used the same window size σ in both $\hat{h}(\mathbf{x})$ and $\hat{g}(\mathbf{x})$. These estimators are asymptotically unbiased and consistent under certain conditions. Proofs are deferred to a more comprehensive version of this paper.

For notational convenience, we denote $u^{\frac{1}{2}}(\mathbf{x}_i) = u_i^{\frac{1}{2}}$. Using these estimators, we then have $\int \hat{h}(\mathbf{x}) \hat{g}(\mathbf{x}) d\mathbf{x}$

$$\begin{aligned}
&= \int \frac{1}{N_1} \sum_{i=1}^{N_1} u_i^{\frac{1}{2}} W_\sigma(\mathbf{x}, \mathbf{x}_i) \frac{1}{N_2} \sum_{j=1}^{N_2} u_j^{\frac{1}{2}} W_\sigma(\mathbf{x}, \mathbf{x}_j) d\mathbf{x} \\
&= \frac{1}{N_1 N_2} \sum_{i,j=1}^{N_1, N_2} u_i^{\frac{1}{2}} u_j^{\frac{1}{2}} \int W_\sigma(\mathbf{x}, \mathbf{x}_i) W_\sigma(\mathbf{x}, \mathbf{x}_j) d\mathbf{x} \\
&= \frac{1}{N_1 N_2} \sum_{i,j=1}^{N_1, N_2} u_i^{\frac{1}{2}} u_j^{\frac{1}{2}} W_{\sqrt{2}\sigma}(\mathbf{x}_i, \mathbf{x}_j),
\end{aligned} \tag{6}$$

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

$$\int \hat{h}^2(\mathbf{x}) d\mathbf{x} = \frac{1}{N_1^2} \sum_{i,i'=1}^{N_1, N_1} u_i^{\frac{1}{2}} u_{i'}^{\frac{1}{2}} W_{\sqrt{2}\sigma}(\mathbf{x}_i, \mathbf{x}_{i'}), \tag{7}$$

and, likewise

$$\int \hat{g}^2(\mathbf{x}) d\mathbf{x} = \frac{1}{N_2^2} \sum_{j,j'=1}^{N_2, N_2} u_j^{\frac{1}{2}} u_{j'}^{\frac{1}{2}} W_{\sqrt{2}\sigma}(\mathbf{x}_j, \mathbf{x}_{j'}). \tag{8}$$

For further notational convenience, we define the $u(\mathbf{x})$-dependent window, or kernel function, $k_{ij_u} = u_i^{\frac{1}{2}} u_j^{\frac{1}{2}} W_{\sqrt{2}\sigma}(\mathbf{x}_i, \mathbf{x}_j)$. Hence, a Parzen window-based estimator for the argument of the logarithm in (3) can be expressed as

$$S(p_1, p_2) = \frac{\frac{1}{N_1 N_2} \sum_{i,j=1}^{N_1, N_2} k_{ij_u}}{\sqrt{\frac{1}{N_1^2} \sum_{i,i'=1}^{N_1, N_1} k_{ii'_u} \frac{1}{N_2^2} \sum_{j,j'=1}^{N_2, N_2} k_{jj'_u}}}. \tag{9}$$

This estimator can be related to a Mercer kernel feature space. The reason for this connection is that $W_{\sqrt{2}\sigma}(\mathbf{x}_i, \mathbf{x}_j)$ is a Gaussian Mercer kernel [21], [5], thus computing an inner-product in some Mercer kernel feature space. It can

be easily shown that $u(\mathbf{x})$ being a non-negative weighting function implies that k_{ij_u} is also a Mercer kernel. Hence, this kernel computes an inner-product in a kernel space, that is $k_{ij_u} = \langle \mathbf{\Phi}_u(\mathbf{x}_i), \mathbf{\Phi}_u(\mathbf{x}_j) \rangle$. Here, $\mathbf{\Phi}_u(\mathbf{x}_i)$ corresponds to the non-linear mapping from the input space to the kernel space, which depends both on the window function W and the weighting function $u(\mathbf{x})$. For simplicity, we denote $\mathbf{\Phi}_{i_u} = \mathbf{\Phi}_u(\mathbf{x}_i)$.

Thus, we may analyze the measure $S(p_1, p_2)$ in terms of inner-products in a Mercer kernel feature space as follows.

$$S(p_1,p_2) = \frac{\frac{1}{N_1 N_2}\sum_{i,j=1}^{N_1,N_2}\langle \mathbf{\Phi}_{i_u}, \mathbf{\Phi}_{j_u}\rangle}{\sqrt{\frac{1}{N_1^2}\sum_{i,i'=1}^{N_1,N_1}\left\langle \mathbf{\Phi}_{i_u}, \mathbf{\Phi}_{i'_u}\right\rangle \frac{1}{N_2^2}\sum_{j,j'=1}^{N_2,N_2}\left\langle \mathbf{\Phi}_{j_u}, \mathbf{\Phi}_{j'_u}\right\rangle}}$$

$$= \frac{\left\langle \frac{1}{N_1}\sum_{i=1}^{N_1}\mathbf{\Phi}_{i_u}, \frac{1}{N_2}\sum_{j=1}^{N_2}\mathbf{\Phi}_{j_u}\right\rangle}{\sqrt{\left\langle \frac{1}{N_1}\sum_{i=1}^{N_1}\mathbf{\Phi}_{i_u}, \frac{1}{N_1}\sum_{i'=1}^{N_1}\mathbf{\Phi}_{i'_u}\right\rangle \left\langle \frac{1}{N_2}\sum_{j=1}^{N_2}\mathbf{\Phi}_{j_u}, \frac{1}{N_2}\sum_{j'=1}^{N_2}\mathbf{\Phi}_{j'_u}\right\rangle}}$$

$$= \frac{\langle \mathbf{m}_{1_u}, \mathbf{m}_{2_u}\rangle}{\sqrt{\langle \mathbf{m}_{1_u}, \mathbf{m}_{1_u}\rangle \langle \mathbf{m}_{2_u}, \mathbf{m}_{2_u}\rangle}} = \cos\angle(\mathbf{m}_{1_u}, \mathbf{m}_{2_u}), \quad (10)$$

where $\mathbf{m}_{1_u} = \frac{1}{N_1}\sum_{i=1}^{N_1}\mathbf{\Phi}_{i_u}$ and $\mathbf{m}_{2_u} = \frac{1}{N_2}\sum_{j=1}^{N_2}\mathbf{\Phi}_{j_u}$ are Mercer kernel feature space *mean vectors* associated with the data points drawn from the pdfs $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$, respectively. Thus, the information theoretic divergence measure between pdfs that we started out with turns out to have a dual expression in a kernel feature space. In that space, the cost function measures *the cosine of the angle between the cluster mean vectors*.

Extended to more than two pdfs, the divergence measure basically corresponds to a measure of the pairwise sum of the *cosine of the angle* between cluster mean vectors in a kernel induced feature space, as

$$S(p_1,\ldots,p_C) = \frac{1}{\kappa}\sum_{i=1}^{C-1}\sum_{j>i}\cos\angle(\mathbf{m}_{i_u}, \mathbf{m}_{j_u}). \quad (11)$$

A. Approximation by the Spectral Properties of the Kernel Matrix

We will now consider the special case that $u(\mathbf{x}) \equiv 1$. In that case, the estimators (5) and (4) reduce to the ordinary Parzen window estimators for the densities $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$. Moreover, the kernel function becomes $k_{ij} = W_{\sqrt{2}\sigma}(\mathbf{x}_i, \mathbf{x}_j)$, the usual Gaussian Mercer kernel. Then, the measure $S(p_1, p_2)$ can be expressed in terms of the *affinity* matrix $\mathbf{K}$. The affinity matrix is defined such that element (i, j) of $\mathbf{K}$ equals k_{ij}. It is well known [22], [23], [24] that the actual mapping $\mathbf{\Phi}(\cdot)$ to the kernel space can be approximated by the eigenspectrum of the matrix $\mathbf{K}$. The eigendecomposition is given by $\mathbf{K} = \mathbf{E}\mathbf{D}\mathbf{E}^T$, where $\mathbf{E}$ is a matrix having the eigenvectors of $\mathbf{K}$ as columns, and $\mathbf{D}$ is a diagonal matrix of corresponding eigenvalues. The mapping is approximated by the C largest eigenvalues and eigenvectors as

$$\mathbf{\Phi}(\mathbf{x}_i) \approx [\sqrt{\tilde{\lambda}_1}e_{1l}, \ldots, \sqrt{\tilde{\lambda}_C}e_{Cl}]^T, \quad (12)$$

where e_{jl} denotes the lth element of the jth eigenvector of $\mathbf{K}$ and $\tilde{\lambda}_j$ is the corresponding eigenvalue, where $\tilde{\lambda}_1 \geq \cdots \geq \tilde{\lambda}_C$). This is the procedure we will follow in order to generate the data set corresponding to the kernel induced feature space.

Similarly, for a general weighting function $u(\mathbf{x})$, the data mapping is approximated by (12) using the eigenspectrum of the kernel matrix $\mathbf{K}_u$, defined such that element (i, j) of $\mathbf{K}_u$ equals k_{ij_u}.

In [8], we considered the particular weighting function $u(\mathbf{x}) = f^{-1}(\mathbf{x})$, where $f(\mathbf{x})$ is the overall pdf of the input data set. We showed that when $f(\mathbf{x})$ is estimated by the Parzen window method, the resulting kernel matrix is in fact the Laplacian matrix $\mathbf{K}_{f^{-1}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{K}\mathbf{D}^{-\frac{1}{2}}$. Hence, in this special case, the data mapping is approximated using the eigenspectrum of the Laplacian matrix.

B. Interpretation of the Weighting Function

Based on our starting point, i.e. the divergence measure between pdfs, we are in a position to provide an analysis of the effect of the two particular weighting functions, $u(\mathbf{x}) = 1$ and $u(\mathbf{x}) = f^{-1}(\mathbf{x})$. As mentioned, these functions correspond to a data mapping associated with the affinity matrix and the Laplacian matrix, respectively.

For $u(\mathbf{x}) = 1$ no region in the input space is weighted more than other regions, in the computation of the inner-product integral, that is $\langle p_1(\mathbf{x}), p_2(\mathbf{x})\rangle = \int p_1(\mathbf{x})p_2(\mathbf{x})u(\mathbf{x})d\mathbf{x} = \int p_1(\mathbf{x})p_2(\mathbf{x})d\mathbf{x}$.

However, for $u(\mathbf{x}) = f^{-1}(\mathbf{x})$, the weighting of the inner-product integral in a particular region is inversely proportional to the probability density function in that region. This means that a data point associated with a low probability region are given a high weight. Conversely, a data point in a region of high probability are given a low weight. For example, data points on the borderline between clusters will be given high weights. This is also the case for data points associated with sparse clusters, i.e. consisting of few and spread data points. This weighting property therefore explains some of the difference between clustering based on the affinity matrix and clustering based on the Laplacian matrix, in terms of inner-products and weighting functions. To our knowledge, this viewpoint is new.

Figure 1 aims to illustrate this point. Two one-dimensional data sets are used to estimate the densities $p_1(x)$ and $p_2(x)$ based on the Parzen window method. The curves corresponding to these estimates are shown in the figure. Also, the overall pdf $f(x)$ has been estimated by the same procedure (solid curve). The relative weighting on each of the data samples, given by $u(x_i) = f^{-1}(x_i)$ are shown as the bars. It can be seen that the data points situated close to each other are designated a low weighting, because they correspond to a high probability region. On the other hand, e.g. the data point

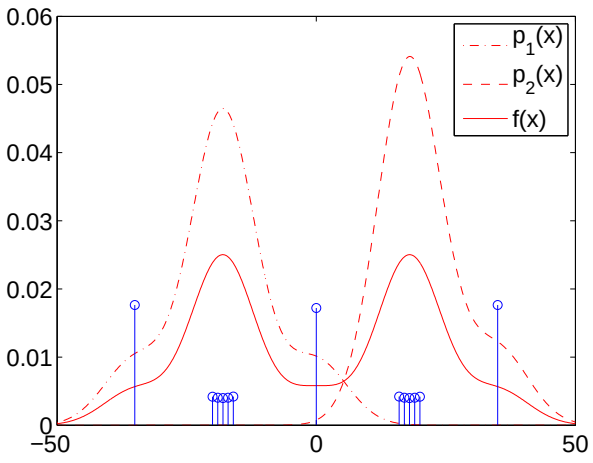


Fig. 1. Illustration of weighting given by $u(\mathbf{x}) = f^{-1}(\mathbf{x})$.

in the middle, corresponding to low probability, is designated a high weight.

III. A Spectral Clustering Algorithm

In [8], we presented some of the theoretical results reviewed in the previous section. No actual clustering algorithm was implemented. In this section, we will show that we are able to derive a spectral clustering algorithm from our well-defined information theoretic cost function. This algorithm is general with respect to the kernel matrix, in the sense that the data mapping can be approximated using the affinity matrix, the Laplacian matrix, or in theory other kernel matrices induced by the function $u(\mathbf{x})$.

We wish to cluster the data points, i.e. assign cluster membership values, such that $S(p_1, \ldots, p_C)$ takes its smallest value, because this corresponds to a maximum value for the divergence $D(p_1, \ldots, p_C)$. Hence, we must assign cluster memberships such that the *angle* between the kernel feature space cluster mean vectors are kept as large as possible. This is a very simple clustering criterion. It can be accomplished simply by measuring the cosine of the angle between a feature space data point and all the feature space mean vectors, for then to assign the data point to the mean vector, or cluster, which corresponds to the largest value.

Such a clustering procedure requires a method for initializing mean vectors in the kernel feature space. We will come back to this shortly.

In pseudo-code, the proposed information theoretic angle-based spectral clustering algorithm consists of the following steps

1) Select an inner-product weighting function $u(\mathbf{x})$.
2) Select a kernel size σ.
3) Construct $\mathbf{K}_u$.
4) Perform the data mapping using (12) and $\mathbf{K}_u$.
5) Initialize mean vectors in the kernel space.
6) For all $\mathbf{x}_t$, $t = 1, \ldots, N$:

$$\mathbf{x}_t \rightarrow \omega_i : \quad \max_i \cos \angle(\mathbf{\Phi}_u(\mathbf{x}_t), \mathbf{m}_{i_u}).$$

7) Update mean vectors.
8) Repeat steps 6-8 until convergence.

One may compute the value of the cost function at each iteration step. If the decrease in the value of the cost function is very small from one iteration to the next, the algorithm has converged and may terminate.

A. Initializing Mean Vectors

The kernel feature space mean vectors may be initialized randomly. However, by performing an "ideal case" analysis of $\mathbf{K}_u$ (clusters are "infinitely" far apart, see e.g. [18]), it can be shown that the "ideal" mean vectors are proportional to $\mathbf{m}_{1_u} = [\pm 1, 0, \ldots, 0]^T$, $\mathbf{m}_{2_u} = [0, \pm 1, \ldots, 0]^T$, and so on. Therefore, if the mean value of the first eigenvector is positive, we initialize $\mathbf{m}_{1_u} = [1, 0, \ldots, 0]^T$. Otherwise $\mathbf{m}_{1_u} = [-1, 0, \ldots, 0]^T$. 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.

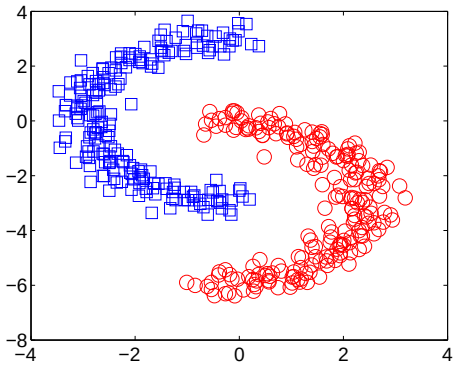
B. A Comment on Kernel Size Selection

As shown in this paper, the Parzen window and the Mercer kernel are equivalent. In theory therefore, data-driven rules for Parzen window size selection known from statistics may be helpful in determining an appropriate Mercer kernel size.

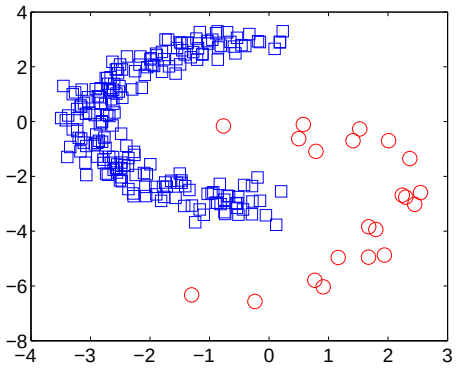
Unbiased least squares cross-validation [25] produces a window size σ_{LSCV} which minimizes the mean squared integrated error (MISE) between the Parzen window estimator and the true density. It is often to small, because it favors unbiasedness, i.e. a small kernel size. By estimating the standard deviation according to a Gaussian density, an expression for the asymptotic optimal MISE kernel size is given by $\sigma_{\text{AMISE}} = \sigma_X \left[\frac{4}{(2d+1)N}\right]^{\frac{1}{d+4}}$, 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 [26]. For low dimensional and non-Gaussian data sets, this kernel size is often too large. One approach may therefore be to use the mean of these values, σ_{mean}. However, the higher the dimensionality of the data sets, the more difficult it gets to determine an appropriate kernel size. This is a problem for all kernel-based algorithms.

IV. Some Clustering Experiments

In the following, we perform some clustering experiments using both the affinity matrix and the Laplacian matrix. We demonstrate that the information theoretic angle-based spectral clustering algorithm may perform reasonably well, meaning that the underlying natural grouping of the data is unraveled. The number of clusters to be discovered is user-specified. Note that there is no random component in this algorithm. This means that the clustering result for a particular data set is always the same even in repeated trials, given the same kernel size in every trial. We present the clustering *error-rate* as a function of the kernel size, over a range of kernel sizes. For comparison, we indicate in each case the kernel sizes determined by data-driven statistical rules.



(a) Dense clusters.



(b) Dense and sparse clusters.

Fig. 2. Crescent-shaped data set, with correct labeling indicated.

We compare with the Ng et al. [18] algorithm. In this method, the input data is mapped to a feature space determined by the eigenvectors corresponding to the C largest eigenvalues of the Laplacian matrix. In that space, the data is normalized to unit length. Thereafter, the C-means algorithm [19] is used for the actual clustering. To make the comparison fair, we initialize mean vectors also for this method using the strategy outlined in section III-A. We also use the same kernel size for all methods.

A. Synthetic Data Sets

Figures 2 (a) and (b) show two data sets having a crescent-shaped structure. The clusters constituting the data set in (a) are both quite dense. The cluster to the left consists of 209 data points, while the cluster to the right consists of 210 data points. In (b), the leftmost cluster is also quite dense, and consists of 209 data points. On the other hand, the rightmost cluster is much more sparse, consisting of only 21 data points. The "correct" labeling is indicated. This information is of course not available to the clustering algorithms.

Figure 3 (a) shows the clustering results for the Fig. 2 (a) data set. For a kernel size less than 0.38, the clustering result based on the affinity matrix is not very good. However, for a kernel size in the range $0.38 \leq \sigma \leq 0.65$, the correct result is obtained. Thereafter, for increasing kernel size, the result gets worse. The clustering result based on the Laplacian matrix is better with respect to the range of σ. A perfect result is obtained also for a very small kernel size. Note that $\sigma_{mean} = 0.5$ while $\sigma_{mise} = 0.8$. Note also that the result obtained by the Ng et al. method is nearly identical to the result obtained by the proposed angle-based algorithm when using the Laplacian matrix.

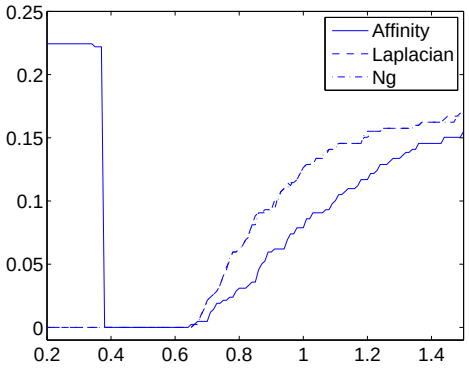
Figure 3 (b) shows the clustering results for the Fig. 2 (b) data set. In this case, the data mapping provided by the affinity matrix is not appropriate in order to obtain reasonable results for any kernel size. It clearly can not handle the sparse cluster. On the other hand, the clustering based on the Laplacian matrix is perfect for small kernel sizes. However, for $\sigma > 0.51$ reasonable results are no longer obtained. But it seems as if the $u(\mathbf{x}) = f^{-1}(\mathbf{x})$ weighting has a positive effect on the clustering results. Again, the Ng et al. method performs equally good.

Figure 3 (c) illustrates the $u(\mathbf{x}) = f^{-1}(\mathbf{x})$ weighting property for these two data sets, using a fixed kernel size $\sigma = 0.5$. The stapled line corresponds to the weighting associated with the Fig. 2 (a) data set. The first 209 data points correspond to the leftmost cluster, while the last 210 data points correspond to the rightmost cluster. All the data points are in this case weighted fairly equally. This is to be expected, since all the data points are situated in regions having more or less the same probability density structure. The solid line indicates the weighting associated with the Fig. 2 (b) data set. The data points corresponding to the dense (leftmost) cluster are weighted equally. However, the data points associated with the sparse (rightmost) cluster are designated higher weights. In some cases much higher weights. These data points will therefore be the most important for the clustering performance. Note that for a "large" kernel size, the pdf estimate will be more smooth than for a "small" kernel size, basically smoothing out the difference in the weighting of the data points. This may explain why the results gets worse for larger kernel sizes also for clustering based on the Laplacian matrix.

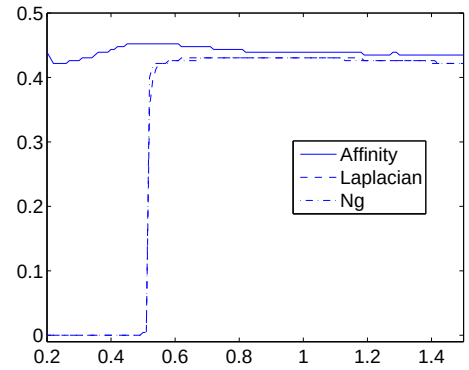
We have also included an experiment using the data set shown in Fig. 4. There are a total of $N = 315$ data patterns. The inner-most circle contains $N_1 = 63$ data points. The radius is so small that it looks almost like a point-cluster, hence it is very dense. The middle circle and the outer circle both contain $N_2 = N_3 = 126$ data patterns. Thus, the outer circle is the most sparse cluster. In this case, clustering based on the affinity matrix does not provide reasonable results. Clustering based on the Laplacian matrix (and the Ng et al. method) provides perfect results for $0.3 \leq \sigma \leq 1.9$. Note that $\sigma_{mean} = 1.8$.

B. Real Data Sets

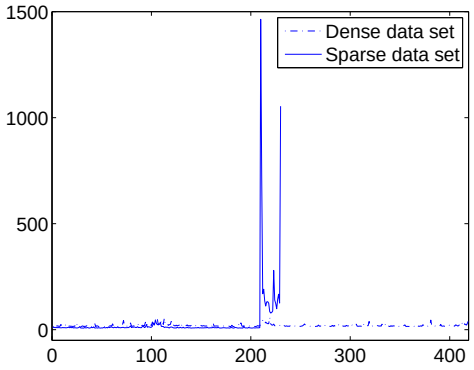
The Wisconsin breast-cancer data set (extracted from the UCI repository [27]) consists of two classes of tumors,



(a) Results for Fig. 2 (a) data set.



(b) Results for Fig. 2 (b) data set.



(c) Illustration of weighting function.

Fig. 3. Clustering of crescent-shaped data sets (rrror-rate vs. kernel size.)

namely *benign* and *malignant*. There are totally 683 data points, where 444 correspond to the benign class and 239 to the malignant class. It is a nine-dimensional dataset with features related to clump thickness, uniformity of cell size

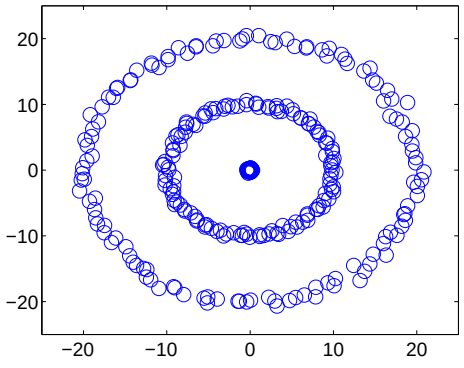


Fig. 4. Clustering of ring-shaped data set. Correct result obtained for a kernel size in the range $0.3 \leq \sigma \leq 1.9$.

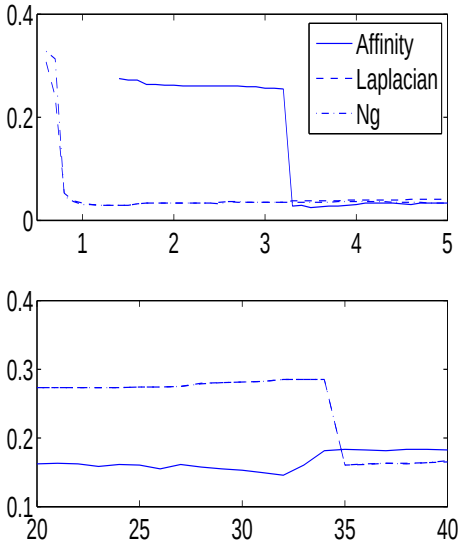
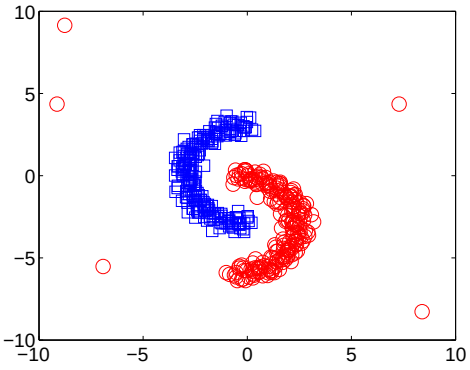


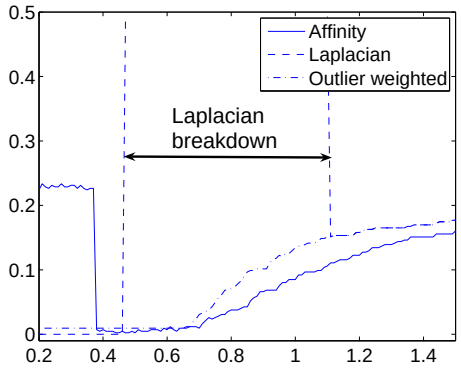
Fig. 5. Clustering of real data sets (error-rate vs. kernel size.). Upper panel: Wisconsin breast-cancer, Lower panel: Pen-based handwritten digit recognition.

and shape, etc. The upper panel of Fig. 5 shows the clustering result as a function of the kernel size. The clustering based on the affinity matrix does not perform well for relatively small kernel sizes. For large kernel sizes, the result is quite good. Based on the Laplacian matrix, the result is good for a wide range of kernel sizes. Here, $\sigma_{mise} = 1.6$.

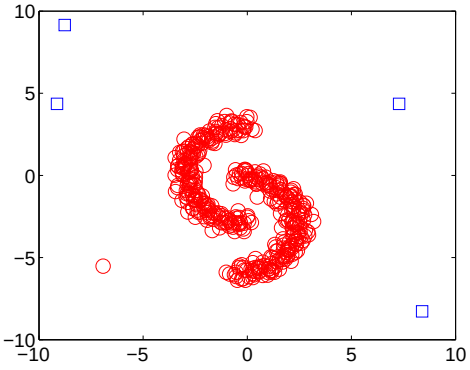
The lower panel of Fig. 5 shows the clustering result of the pendigits data set (also extracted from UCI). This data set was created for pen-based handwritten digit recognition. The data set is 16-dimensional. All attributes are integers in the range $[0, 100]$. We use data vectors corresponding to the digits 0, 1 and 2. These clusters consist of 363, 364 and 364 data patterns, respectively. Perhaps surprisingly, in this case the clustering based on the affinity matrix provides the best result for the smaller kernel sizes in the range shown ($\sigma_{mise} = 20$). For smaller kernel sizes, the result based on



(a) True labels, data set with outliers.



(b) Clustering results (error-rate) as a function of kernel size.



(c) Solution where the Laplacian method breaks down.

Fig. 6. Illustrating the problem of high weighting of outliers using the Laplacian matrix for clustering.

the Laplacian matrix (and the Ng et al. method) is not that good. For the larger kernel sizes, the result based on the Laplacian matrix is a little bit better.

We do not at present have a good explanation as to why these results are observed on the pendigits data set. They may be related to the outlier-problem we will discuss in the next section. First, we will comment on why the angle-based clustering using the Laplacian matrix and the Ng et al. method performs almost identically good.

C. Comment on a Connection to the Ng et al. Method

In [8], we argued that spectral clustering based on the Cauchy-Schwarz divergence using the Laplacian matrix, and clustering using the Ng et al. [18] method, would probably give quite similar results. Indeed, the experiments conducted in this paper seem to be confirmatory in that respect. First, note that we incorporate the eigenvalues in addition to the eigenvectors in the data mapping, in contrast to the Ng et al. method which uses only the eigenvectors. However, it is often the case that the C largest eigenvalues are quite close in range. Thus, the inclusion of the eigenvalues in many cases does not have much effect on the mapping. Second, the normalization of the data followed by C-means clustering basically corresponds to clustering based on an angular measure. Hence, it can be seen that the heuristic Ng et al. algorithm in effect achieves the same goal as the more theoretically well-defined information theoretic clustering.

V. Outlier Robustness by Choice of Weighting Function

On the experiments conducted so far, it seems as if the weighting given by $u(\mathbf{x}) = f^{-1}(\mathbf{x})$ for the most part has a positive effect. However, we have observed that this is not necessarily always the case. The reason is that outliers in the data set will also be given high weights using this weighting. In fact, even if there is only one single outlier in the data set, this particular data point may be designated the highest weight of all the data points. Figure 6 (a) shows the same data set as in Fig. 2 (a), only that now a few outliers are included. These outliers don't really belong to any of the clusters, although in the figure they are marked with the circle-symbol. Figure 6 (b) shows the clustering results. For a wide range of kernel sizes, the clustering based on the Laplacian matrix (also the Ng et al. method) brakes completely down. A typical result is shown in Figure 6 (c). The reason is that the outliers become dominant because of the weighting. In fact, in such a situation, clustering based on the affinity matrix is more robust in this particular kernel size range. For very small kernel sizes, the outlier problem is less, since the affinity between any data point and an outlier is so small anyway.

As a preliminary experiment, we have tried to actively use the weighting function $u(\mathbf{x})$ to avoid this problem. The approach taken is to try to identify possible outliers. Instead of giving these data points huge weights, we assign them small weights. An outlier is identified as follows: If a data point has no neighbors within a 3σ neighborhood, then it must be very isolated in the input data space, and hence defined as an outlier. In the preliminary experiment, $u(\mathbf{x}_i) = f^{-1}(\mathbf{x}_i)$ for all data points except the outliers.

For an outlier, we designate a weight which correspond to 0.01 times $W_{\sqrt{2}\sigma}(0)$. We call such an $u(\mathbf{x})$ *outlier weighted*. In Figure 6 (b) it can be seen that the outlier weighted clustering performs equally good as the Laplacian-based for small kernel sizes and large kernel sizes, but it *does not break down* in the critical kernel size range. We consider this preliminary experiment promising. We intend to investigate more closely alternative weighting schemes in future work.

VI. Conclusions

In this paper, we have analyzed an information theoretic cost function, and shown that it has a dual expression as an angular measure in a Mercer kernel feature space. The effect of the weighting function has been analyzed for the cases $u(\mathbf{x}) = 1$ and $u(\mathbf{x}) = f^{-1}(\mathbf{x})$. An information theoretic angle-based spectral clustering algorithm has been derived an tested on synthetic and real data. It has been shown to be able to unravel the natural grouping of the data. The differences between clustering based on the affinity matrix and the Laplacian matrix has been emphasized. The similarity to the Ng et al. clustering algorithm has been commented upon. We have also discussed the problem of outliers, and shown that such data points may receive such a high weighting that clustering based on the Laplacian matrix may break down. In a preliminary experiment, we have indicated that the weighting function may be actively used in order to limit this problem.

References

[1] J. Principe, D. Xu, and J. Fisher, "Information Theoretic Learning," in *Unsupervised Adaptive Filtering*, S. Haykin (Ed.), John Wiley & Sons, New York, 2000, vol. I, Chapter 7.

[2] D. Erdogmus, *Information Theoretic Learning: Renyi's Entropy and its Applications to Adaptive Systems Training*, Ph.D. thesis, University of Florida, Gainesville, FL, USA, 2002.

[3] 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*, vol. 51, no. 7, pp. 1966–1978, 2003.

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

[5] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An Introduction to Kernel-Based Learning Algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, 2001.

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

[7] 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*, Sao Luis, Brazil, September 29 - October 1, 2004, pp. 93–102.

[8] 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, Cambridge, 2005, pp. 625–632.

[9] A. Renyi, "On Measures of Entropy and Information," *Selected Papers of Alfred Renyi, Akademiai Kiado, Budapest*, vol. 2, pp. 565–580, 1976.

[10] E. Parzen, "On the Estimation of a Probability Density Function and the Mode," *The Annals of Mathematical Statistics*, vol. 32, pp. 1065–1076, 1962.

[11] M. Fiedler, "Algebraic Connectivity in Graphs," *Czechoslovak Mathematics Journal*, vol. 23, pp. 298–305, 1973.

[12] 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*, San Jose, USA, November 29 - December 2, 2001, pp. 107–114.

[13] 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.

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

[15] 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*, Santa Clara, USA, November 11-14, 1991, pp. 10–13.

[16] A. Pothen, H. D. Simon, and K. P. Liou, "Partitioning Sparse Matrices with Eigenvectors of Graphs," *SIAM Journal of Matrix Analysis and Applications*, vol. 11, no. 3, pp. 430–452, 1990.

[17] 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.

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

[19] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *Proceedings of Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, 1967, pp. 281–297.

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

[21] 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.

[22] 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.

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

[24] Y. Bengio, P. Vincent, and J.-F. Paiement, "Spectral Clustering and Kernel PCA are Learning Eigenfunctions," Tech. Rep., Département d'informatique et recherche opérationelle, université de Montréal, Montréal, Canada, 2003.

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

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

[27] R. Murphy and D. Ada, "UCI Repository of Machine Learning databases," Tech. Rep., Dept. Comput. Sci. Univ. California, Irvine, 1994.