Information Cut for Clustering using a Gradient Descent Approach

Robert Jenssen*[1], Deniz Erdogmus⋆, Kenneth E. Hild II†, Jose C. Principe‡ and Torbjørn Eltoft*,

*Department of Physics, University of Tromsø, N-9037 Tromsø, Norway
⋆Oregon Graduate Institute, OHSU, Portland OR. 97006, USA
†Biomagnetic Imaging Lab, University of California, SF, CA. 94143, USA
‡Computational NeuroEngineering Laboratory, Department of Electrical and Computer Engineering, University of Florida, FL. 32611, USA

Abstract

We introduce a new graph cut for clustering which we call the Information Cut. It is derived using Parzen windowing to estimate an information theoretic distance measure between probability density functions. We propose to optimize the Information Cut using a gradient descent-based approach. Our algorithm has several advantages compared to many other graph-based methods in terms of determining an appropriate affinity measure, computational complexity, memory requirements and coping with different data scales. We show that our method may produce clustering and image segmentation results comparable or better than the state-of-the art graph-based methods.

Key words: Graph theoretic cut, information theory, Parzen window density estimation, clustering, gradient descent optimization, annealing.

1 Introduction

In signal processing and data analysis, it is often desirable to partition, or cluster, a data set into subsets. Several textbooks provide surveys of traditional clustering techniques, see e.g. [1–3].

Recently, a new line of research in clustering has emerged. It is based on the notion of a graph *cut*. A set of points, $\mathbf{x}_l, l = 1, \ldots, N$, in an arbitrary data space can be represented as a weighted undirected graph $\mathcal{G}$. Each node in the graph corresponds

[1] Corresponding author. Tel. (+47) 776 46493, Fax. (+47) 776 45580, Email: robertj@phys.uit.no, Web: www.phys.uit.no/∼robertj.

to a data point. The edge formed between a pair of nodes, say l and l', is weighted according to the similarity between the corresponding data points. The edge-weight is denoted $k_{ll'}$. The graph *cut* provides a measure of the cost of partitioning a graph $\mathcal{G}$ into two subgraphs $\mathcal{G}_1$ and $\mathcal{G}_2$, and is defined as

$$Cut(\mathcal{G}_1, \mathcal{G}_2) = \sum_{i,j=1}^{N_1,N_2} k_{ij}, \tag{1}$$

where the index $i = 1, \ldots, N_1$, runs over the N_1 nodes of subgraph $\mathcal{G}_1$ and the index $j = 1, \ldots, N_2$, runs over the N_2 nodes of subgraph $\mathcal{G}_2$. That is, the *cut* measures the weight of the edges which have to be removed in order to create the two subgraphs. Wu and Leahy [4] first proposed to minimize the *cut*-cost as a means for clustering and image segmentation.

Shi and Malik [5] pointed out that the *cut* tends to produce a skewed data partition. It will in fact be minimized if one node in the graph is isolated in one group, and all the rest in the other group. They proposed the heuristically motivated Normalized Cut (NC), defined as

$$NC(\mathcal{G}_1, \mathcal{G}_2) = \frac{Cut(\mathcal{G}_1, \mathcal{G}_2)}{Assoc(\mathcal{G}_1, \mathcal{G})} + \frac{Cut(\mathcal{G}_1, \mathcal{G}_2)}{Assoc(\mathcal{G}_2, \mathcal{G})}, \tag{2}$$

where $Assoc(\mathcal{G}_1, \mathcal{G}) = \sum_{i,l=1}^{N_1,N} k_{il}$ and $Assoc(\mathcal{G}_2, \mathcal{G}) = \sum_{j,l=1}^{N_2,N} k_{jl}$, i.e. the total connection from nodes in $\mathcal{G}_1$ ($\mathcal{G}_2$) to all nodes in the graph $\mathcal{G}$. Shi and Malik optimized the Normalized Cut based on the eigenvectors of the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{K}$. Here, $\mathbf{D}$ is a diagonal matrix where the mth diagonal entry is given by $d_m = \sum_{l=1}^{N} k_{ml}$. The matrix $\mathbf{K} = k_{ll'}, l = 1, \ldots, N,\ l' = 1, \ldots, N$, is called the *affinity* matrix.

Several other heuristically motivated *cut* normalizations have also been proposed, such as the min-max cut [6], the typical cut [7] and the BCut [8]. When the optimization is carried out based on the eigendecomposition (spectrum) of a matrix, the methods are referred to as graph spectral clustering methods. Graph spectral clustering methods have been promising compared to traditional clustering methods. Other examples of spectral clustering methods can be found in [9–16].

The main problems associated with graph spectral clustering methods are the following. An appropriate affinity measure (edge-weight) must be selected. Often, this corresponds to selecting the width of an exponential kernel function. There is no widely accepted procedure to select this parameter, even though it heavily affects the clustering result. Furthermore, the $(N \times N)$ matrix $\mathbf{K}$ needs to be stored in memory, and possibly other matrices too. In addition, a matrix eigendecomposition needs to be done. The computational complexity of computing an eigenvector of a $(N \times N)$ matrix is in the order of $O(N^2)$. Hence, finding all the eigenvectors scales as $O(N^3)$. Also, it is a concern that the various graph spectral cost functions are based on heuristics, and lack a clear theoretical foundation.

In this paper, we introduce a new theoretically well-defined graph cut for clus-

tering, named the Information Cut. The Information Cut is basically a Parzen window-based estimator for the information theoretic Cauchy-Schwarz divergence measure between probability density functions. We are faced with the task of assigining cluster memberships to the data points such that the Information Cut is minimized. We propose a gradient-based optimization strategy, as opposed to an eigenvector-based approach. There are several advantages to our approach. We are able to select an *appropriate affinity measure* (kernel size) based on *data-driven* rules for optimal Parzen window density estimation. Furthermore, there is no need to store the affinity matrix in the computer memory. And also, using a stochastic sampling approach to gradient estimation, our resulting clustering algorithm has a relatively moderate computational complexity of $O(MN)$, $M \ll N$, at each iteration cyclus. Here, M is the number of stochastically selected samples to be used in the computation. The main disadvantage of a gradient-based approach for optimizing non-convex cost functions is the problem of convergence to a local minimum of the cost function landscape. We incorporate a strategy to reduce this shortcoming by allowing the kernel size to be *annealed* over time. The annealing procedure comes with the benefit that is may help cope with clusters of significantly different scales, by discovering large-scale data structures in the early stages of the algorithm, followed by a "fine-tuning" as the kernel size decreases. We show experimentally that we obtain clustering results which are comparable or better than the state-of-the art graph spectral methods.

Of course, clustering based on information theoretic ideas is not new. However, the coupling between graph-based clustering methods, Parzen windowing and information theory which we present in this paper is new. Our novel algorithm is a substantial improvement over a related algorithm proposed by Gockay and Principe [17], based on Renyi's entropy. Their clustering technique was based on calculating the cost function for all clustering possibilities, hence impractical for anything but very small data sets. Other information theoretic methods include Watanabe [18], who used a coalescence model and a cohesion method to aggregate and shrink the data into desired clusters. Rose et al. [19] employed the robustness properties of maximum entropy inference for vector quantization, and Hofmann and Buhmann [20] applied the same criterion for pairwise clustering. Roberts et al. [21] proposed a clustering method based on minimizing the partition entropy. Recently, Tishby and Slonim [22] proposed the information bottleneck method.

The remainder of this paper is organized as follows. In section 2, the theory behind the Information Cut is derived. In section, 3, a gradient descent-based optimization strategy is outlined. We present some clustering results in section 4. Finally, we make our concluding remarks in section 5 [2].

[2] The connection between information theory and graph theory was first mentioned in [23]. A previous version of the proposed clustering algorithm was introduced by Jenssen et al. in [24].

2 The Information Cut

The Cauchy-Schwarz divergence is given by [25]

$$D_{CS}(p_1, p_2) = -\log \frac{\int p_1(\mathbf{x}) p_2(\mathbf{x}) d\mathbf{x}}{\sqrt{\int p_1^2(\mathbf{x}) d\mathbf{x} \int p_2^2(\mathbf{x}) d\mathbf{x}}}. \tag{3}$$

This is a symmetric measure, such that $0 \leq D_{CS} < \infty$, where the minimum is obtained if and only if $p_1(\mathbf{x}) = p_2(\mathbf{x})$. Since the logarithm is a monotonic function, we may just as well focus on the argument of this function. If the "distance" between the densities is large, the argument of this function will be small. Let us estimate this quantity by replacing the actual pdfs by their Parzen window estimators. Let $\mathbf{x}_i,\ i = 1, \ldots, N_1$, be data points drawn from the density $p_1(\mathbf{x})$, and let $\mathbf{x}_j,\ j = 1, \ldots, N_2$, be data points drawn from $p_2(\mathbf{x})$. Then, the Parzen window estimators for these distributions are [26]

$$\hat{p}_1(\mathbf{x}) = \frac{1}{N_1} \sum_{i=1}^{N_1} W_\sigma(\mathbf{x}, \mathbf{x}_i), \tag{4}$$

$\hat{p}_2(\mathbf{x}) = \frac{1}{N_2} \sum_{j=1}^{N_2} W_\sigma(\mathbf{x}, \mathbf{x}_j)$, 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. In that case,

$$W_\sigma(\mathbf{x}, \mathbf{x}_l) = \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} \exp\left\{ -\frac{||\mathbf{x} - \mathbf{x}_l||^2}{2\sigma^2} \right\}, \tag{5}$$

where $\mathbf{x}_l$ is some data point in the data set. Notice that the assumption of Gaussian Parzen windows is not critical for the following derivation, as shown in Appendix A.

According to the convolution theorem for Gaussian functions, the following relation holds

$$\int W_\sigma(\mathbf{x}, \mathbf{x}_l) W_\sigma(\mathbf{x}, \mathbf{x}_{l'}) d\mathbf{x} = W_{\sqrt{2}\sigma}(\mathbf{x}_l, \mathbf{x}_{l'}). \tag{6}$$

In the remainder of this paper, we denote $W_{\sqrt{2}\sigma}(\mathbf{x}_l, \mathbf{x}_{l'})$ by $k_{ll'}$. Thus, when we replace the actual densities in the argument of (3) by the Parzen window estimators, and utilize (6), we obtain

$$\int \hat{p}_1(\mathbf{x}) \hat{p}_2(\mathbf{x}) d\mathbf{x} = \frac{1}{N_1 N_2} \sum_{i,j=1}^{N_1, N_2} \int W_\sigma(\mathbf{x}, \mathbf{x}_i) W_\sigma(\mathbf{x}, \mathbf{x}_j) d\mathbf{x}$$

$$= \frac{1}{N_1 N_2} \sum_{i,j=1}^{N_1,N_2} k_{ij}. \tag{7}$$

Notice that this expression can be related to the graph-*cut*, by relating the data samples to nodes in a graph. Hence, we relate the samples corresponding to $p_1(\mathbf{x})$ with a graph $\mathcal{G}_1$, and the samples corresponding to $p_2(\mathbf{x})$ with a graph $\mathcal{G}_2$.

Now we perform an exactly similar calculation for the two quantities in the denominator of (3), yielding $\int \hat{p}_1^2(\mathbf{x})d\mathbf{x} = \frac{1}{N_1^2}\sum_{i,i'=1}^{N_1,N_1} k_{ii'}$ and likewise $\int \hat{p}_2^2(\mathbf{x})d\mathbf{x} = \frac{1}{N_2^2}\sum_{j,j'=1}^{N_2,N_2} k_{jj'}$. Based on these expressions, we define the new graph partitioning cost function which we call the Information Cut (IC), as

$$IC(\mathcal{G}_1, \mathcal{G}_2) = \frac{\sum_{i,j=1}^{N_1,N_2} k_{ij}}{\sqrt{\sum_{i,i'=1}^{N_1,N_1} k_{ii'} \sum_{j,j'=1}^{N_2,N_2} k_{jj'}}}. \tag{8}$$

In graph theory, a quantity known as the *volume* of a graph is given by the sum of all the edge-weights in the graph. Hence, $Vol(\mathcal{G}_1) = \sum_{i,i'=1}^{N_1,N_1} k_{ii'}$ and $Vol(\mathcal{G}_2) = \sum_{j,j'=1}^{N_2,N_2} k_{jj'}$. Therefore, the Information Cut may also be written as

$$IC(\mathcal{G}_1, \mathcal{G}_2) = \frac{Cut(\mathcal{G}_1, \mathcal{G}_2)}{\sqrt{Vol(\mathcal{G}_1)Vol(\mathcal{G}_2)}}. \tag{9}$$

In order for the Information Cut to take a small value, there is a trade-off between a small *cut*-value, and a large value for the product of the volumes. Hence, our derivation has introduced a theoretically well-defined normalization which will prevent the Information Cut from obtaining a minimum when one node is isolated from the rest. In the case of partitioning a graph into more than two subgraphs, i.e. subgraphs $\mathcal{G}_c,\ c = 1, \ldots, C$, we define the following multi-way cut

$$IC(\mathcal{G}_1, \ldots, \mathcal{G}_C) = \frac{Cut(\mathcal{G}_1, \ldots, \mathcal{G}_C)}{\sqrt{\prod_{c=1}^{C} Vol(\mathcal{G}_c)}}. \tag{10}$$

where $Cut(\mathcal{G}_1, \ldots, \mathcal{G}_C)$ is the sum of all the edge-weights that need to be removed in order to create C subgraphs.

2.1 Kernel Size Selection Based on Parzen Windowing

In order to derive the Information Cut, we have explicitly used the Parzen window technique for density estimation. In fact, *the Parzen window defines the affinity measure $k_{ll'}$ between two graph nodes l and l'*, given by (6). Parzen kernel size selection has been thoroughly studied in the statistics literature [27–29]. The optimal kernel size may be selected in order to minimize the asymptotic mean integrated

squared error (AMISE) between $\hat{p}(\mathbf{x})$ and the target density $p(\mathbf{x})$. A rough estimate of the AMISE optimal kernel size for d-dimensional data is given by Silverman's rule [27]

$$\sigma_{AMISE} = \sigma_X \left[\frac{4}{(2d+1)N} \right]^{\frac{1}{d+4}}, \tag{11}$$

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. There are also other more advanced approaches to kernel size selection. We use Silverman's rule in our algorithm as an initial value.

3 Clustering by Information Cut Minimization using the Method of Lagrange Multipliers

The cluster membership vectors $\mathbf{m}_i,\ i = 1, \ldots, N$, are defined as C dimensional binary vectors. Only the c'th element of any $\mathbf{m}_i$ equals one, meaning that data pattern $\mathbf{x}_i$ is assigned to cluster c (crisp cluster memberships). We rewrite (8) as a function of the memberships, obtaining

$$IC(\mathbf{m}_1, \ldots, \mathbf{m}_N) = \frac{\frac{1}{2} \sum_{i,j=1}^{N,N} \left(1 - \mathbf{m}_i^T \mathbf{m}_j\right) k_{ij}}{\sqrt{\prod_{c=1}^{C} \sum_{i,j=1}^{N,N} m_{ic} m_{jc} k_{ij}}}. \tag{12}$$

Our goal in clustering is to *assign memberships such that* $IC(\mathbf{m}_1, \ldots, \mathbf{m}_N)$ *is minimized*, because this corresponds to the Cauchy-Schwarz divergence between the corresponding Parzen window estimated pdfs being maximized.

We propose to solve this minimization problem by the method of Lagrange multipliers [30]. Hence, we need to fuzzyfy the membership vectors (each data point may be assigned to several clusters at the same time), similarly to the fuzzy C-means algorithm [31]. .

Let $\mathbf{m}_i \in [0,1]^d$, $i = 1, \ldots, N$. Now we define the following constrained optimization problem:

$$\min_{\mathbf{m}_1, \ldots, \mathbf{m}_N} IC(\mathbf{m}_1, \ldots, \mathbf{m}_N), \tag{13}$$

subject to $\mathbf{m}_j^T \mathbf{1} - 1 = 0, \quad j = 1, \ldots, N$, where $\mathbf{1}$ is a C-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, and derive a fixed-point, gradient-based, learning rule for clustering. Let $m_{ic} = v_{ic}^2,\ c = 1, \ldots, C$. Consider

$$\min_{\mathbf{v}_1, \ldots, \mathbf{v}_N} \ IC(\mathbf{v}_1, \ldots, \mathbf{v}_N), \tag{14}$$

subject to $\mathbf{v}_j^T \mathbf{v}_j - 1 = 0, \quad j = 1, \ldots, N$. The constraints in (14) are equivalent to the constraints in (13). The optimization problem, (14), amounts to adapting the vectors $\mathbf{v}_i,\ i = 1, \ldots, N$, such that

$$\frac{\partial IC}{\partial \mathbf{v}_i} = \left(\frac{\partial IC}{\partial \mathbf{m}_i}^T \frac{\partial \mathbf{m}_i}{\partial \mathbf{v}_i} \right)^T = \mathbf{\Gamma} \frac{\partial IC}{\partial \mathbf{m}_i} \rightarrow \mathbf{0}, \tag{15}$$

where $\mathbf{\Gamma} = diag(2\sqrt{m_{i1}}, \ldots, 2\sqrt{m_{iC}})$. Notice that if all of the diagonal elements $2\sqrt{m_{ic}},\ c = 1, \ldots, C$, are positive, $\frac{\partial IC}{\partial \mathbf{v}_i} \rightarrow \mathbf{0}$ implies that $\frac{\partial IC}{\partial \mathbf{m}_i} \rightarrow \mathbf{0}$. 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. See Appendix B for the derivation of $\frac{\partial IC}{\partial \mathbf{m}_i}$.

The necessary conditions that the solution of (14) must obey, are commonly generated by constructing the Lagrange function [30], given by

$$L = IC(\mathbf{v}_1, \ldots, \mathbf{v}_N) + \sum_{j=1}^{N} \lambda_j \left(\mathbf{v}_j^T \mathbf{v}_j - 1 \right), \tag{16}$$

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, (14), are given by

$$\frac{\partial L}{\partial \mathbf{v}_i} = \frac{\partial IC}{\partial \mathbf{v}_i} + \sum_{k=1}^{N} \lambda_k \frac{\partial}{\partial \mathbf{v}_i} \left(\mathbf{v}_k^T \mathbf{v}_k - 1 \right) = \mathbf{0}, \tag{17}$$

$$\frac{\partial L}{\partial \lambda_j} = \mathbf{v}_j^T \mathbf{v}_j - 1 = 0, \tag{18}$$

for $i = 1, \ldots, N$ and $j = 1, \ldots, N$. From (17) we derive the following *fixed-point* adaption rule for the vector $\mathbf{v}_i$ as follows

$$\frac{\partial IC}{\partial \mathbf{v}_i} + 2\lambda_i \mathbf{v}_i = \mathbf{0} \Rightarrow \mathbf{v}_i^+ = -\frac{1}{2\lambda_i} \frac{\partial IC}{\partial \mathbf{v}_i}, \tag{19}$$

$i = 1, \ldots, N$, 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 (18) as follows

$$
\begin{aligned}
&\mathbf{v}_i^{+T}\,\mathbf{v}_i^{+} - 1 = 0, \\
&\Rightarrow\ \left(-\frac{1}{2\lambda_i}\,\frac{\partial IC}{\partial \mathbf{v}_i}\right)^T \left(-\frac{1}{2\lambda_i}\,\frac{\partial IC}{\partial \mathbf{v}_i}\right) - 1 = 0, \\
&\Rightarrow\ \lambda_i = \frac{1}{2}\sqrt{\frac{\partial IC}{\partial \mathbf{v}_i}^T \frac{\partial IC}{\partial \mathbf{v}_i}}.
\end{aligned} \tag{20}
$$

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.

Notice that the gradient-based optimization technique we have derived does not need for the affinity matrix to be pre-computed and stored in the computer memory.

3.1 Membership Initialization

We initialize the membership vectors randomly according to a uniform distribution. Better initialization schemes may be derived, although in our experiments, this random initialization yields good results. One may also use the output of a different clustering algorithm, such as C-means [32], as the initial cluster memberships. However, this may initialize the algorithm in a local minima, and we have observed that it does not always perform well.

3.2 Kernel Size Annealing

We show experimentally that in our algorithm the convergence problem can to a certain degree be remedied, by allowing the size of the kernel to be annealed over an interval of values around the optimal value. The effect of using a "large" kernel size in the Parzen estimator, is that the pdf estimate will be an over-smoothed version of the actual pdf. Hence, the Information Cut using a "large" kernel is likely to be a smooth function of the memberships, as opposed to using a "small" kernel. The approach taken, is to let the algorithm iterate toward the minimum of the over-smoothed cost function, while continuously decreasing the kernel size, hence leading the algorithm toward the actual global minimum. As opposed to most graph-based clustering algorithms, the annealing procedure therefore has the effect that the *affinity measure will not be fixed, but will start out large, and decrease towards a small value.*

3.3 Reducing Complexity

The computation of all the gradients $\frac{\partial IC}{\partial \mathbf{m}_i}$, $i = 1, \ldots, N$, is an $O(N^2)$ procedure at each iteration. Thus, it is important to reduce the complexity of the algorithm. The

expression for the gradient $\frac{\partial IC}{\partial \mathbf{m}_i}$ is derived in Appendix B. Note that we can calculate all quantities of interest in (24), by determining (25), for $\forall i$. To reduce complexity, we estimate (25) by *stochastically sampling* the membership space, and utilize M randomly selected membership vectors, and corresponding data points, to compute $-\sum_{m=1}^{M} \mathbf{m}_m k_{im}$ as an approximation to (25). 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. $M = 0.2N$.

4 Clustering Experiments

In this section we report some clustering experiments using the proposed Information Cut clustering method. In all experiments, we determine the Information Cut scale parameter using (11), and the number of stochastically selected membership vectors for gradient computation is determined by $M = 0.2N$.

We compare with the Normalized Cut algorithm [5], which is considered by many authors to be a state-of-the-art graph-based clustering method. In [5], the Normalized Cut scale parameter was recommended to be in the order of $10 - 20\%$ of the total range of the Euclidean distances between the feature vectors. We use 15% in our experiments [3].

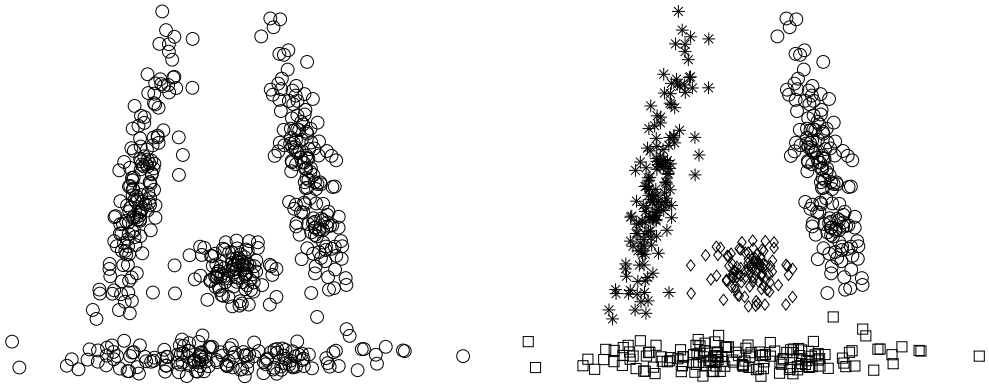
We manually select the number of clusters to be discovered. This is of course a shortcoming compared to a fully automatic clustering procedure, but it is commonly the case in most graph-based clustering algorithms.

We also normalize the variance in each feature vector dimension to one in all experiments (for both algorithms) to avoid problems in case the data scales are significantly different for each feature. We do this since both methods assume a spherical affinity measure.

4.1 Demonstrating the Annealing Property

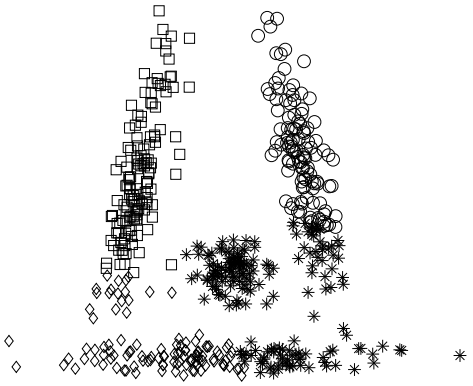
Figure 1 (a) shows a two-dimensional data set consisting of 550 data points. We provide the number of clusters, $C = 4$, as an input parameter to both algorithms. The Parzen window size used in the Information Cut algorithm is determined to be $\sigma = 0.12$ by (11). This means that the effective kernel size used to calculate affinities between data points (nodes) is equal to $\tilde{\sigma} = \sqrt{2}\sigma = 0.17$ by (6). Figure 1 (b) shows a typical result obtained by the Information Cut algorithm. By visual inspection, it clearly makes sense, meaning that the cluster structure underlying the data set seems to have been discovered. Figure 1 (c) shows a typical result obtained by the Normalized Cut algorithm. It is significantly different from the

[3] The Matlab code we use is downloaded from Jianbo Shi's web-page `http://www.cis.upenn.edu/~jshi/GraphTutorial/`.



(a) Two-dimensional data set

(b) Information Cut

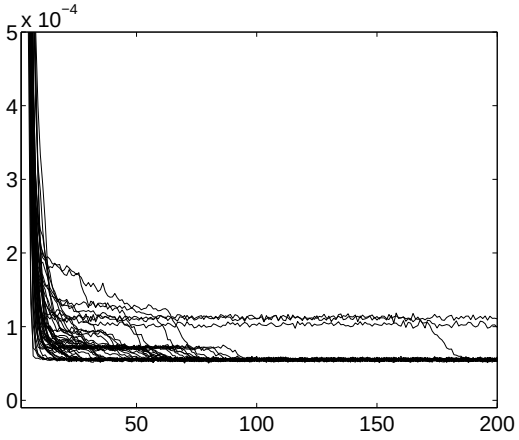


(c) Normalized Cut

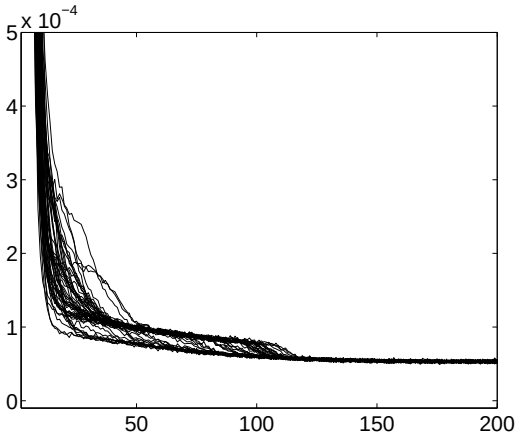
Fig. 1. Original data set shown in (a). Typical (70% of the trials) Information Cut clustering result shown in (b). Typical Normalized Cut clustering result shown in (c).

results obtained by the Information Cut algorithm, and seems to fail to discover the cluster structure. For this particular data set, the scale parameter used in the Normalized Cut algorithm is determined to be $\sigma_{NC} = 0.18$. This means that the edge-weights (affinities) between nodes are roughly the same for both methods. Hence, the significantly different clustering results observed must be a consequence of 1) different clustering cost functions, 2) different optimization techniques, or both.

In the experiment, we had the Information Cut algorithm iterate over 200 iterations. Using a fixed kernel size $\sigma = 0.12$, the result shown in Fig. 1 (b) was obtained in 70% of the 50 clustering trials. In the remaining trials, the algorithm did not converge to a reasonable solution, but rather a solution like the one obtained by the Normalized Cut algorithm. This illustrates the problem of using gradient descent to optimize non-convex cost functions, i.e. convergence to a local minimum. Figure 2 (a) shows a plot where we monitor the value of the Information Cut over the 200 iterations. The plot shows that in many cases (70%) the algorithm converges quickly (oftentimes



(a) Kernel fixed



(b) Kernel annealed

Fig. 2. (a) Using a fixed kernel size, the algorithm does not always converge to the global optimum. The algorithm does not produce satisfying results in 30% of the trials in this case. (b) By annealing the kernel size, the algorithm converges in all trials.

in less than 20 iterations) to a low value. These trials correspond to the clustering result shown in Fig. 1 (b). But sometimes, the algorithm converges to a high value.

Figure 2 (b) shows the value of the Information Cut over 50 clustering trials, where the algorithm operates in *annealing* mode. This means that the kernel size initially has a relatively large value, which is decreased with each iteration. In this particular experiment we anneal the kernel size linearly from a starting value given by $\sigma_{start} = 2\sigma$ to a final value given by $\sigma_{stop} = 0.5\sigma$ over 200 iterations. In this case, the algorithm always converges to a low value for the Information Cut, corresponding to the clustering result shown in Fig. 1 (b). Hence, *the global minimum is obtained in every single trial.* The drawback is of course that several free parameters now must

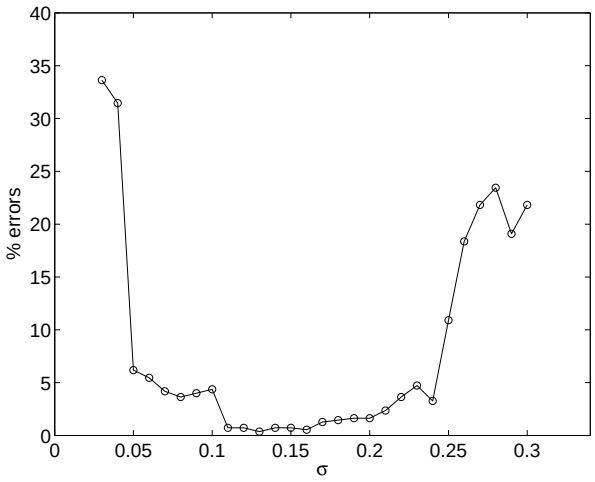
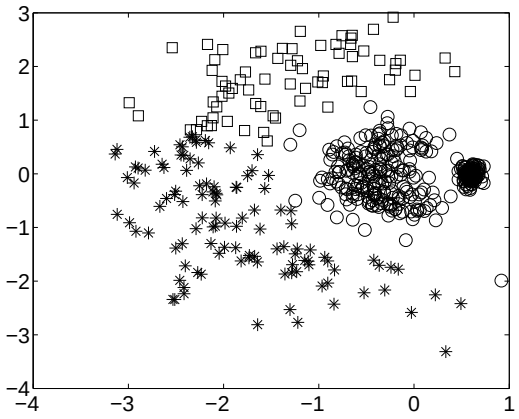


Fig. 3. Clustering result (error percentage compared to the result shown in Fig. 1 (b)) obtained by the Information Cut algorithm over a range of kernel sizes. The "optimal" kernel size, $\sigma = 0.12$ by (11), lies in the middle of the range corresponding to a low error percentage.

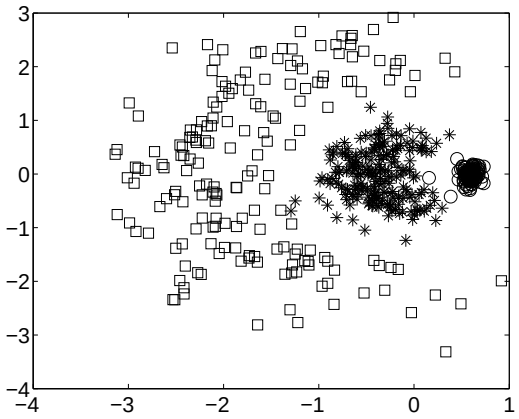
be chosen. Recall that the Information Cut algorithm in fixed kernel mode has no free parameters for the user to choose (except the number of clusters). In annealing mode, the annealing scheme must be chosen. Hence, an upper limit for the kernel size must be selected, a lower limit, and the decay rate. If these parameters are not chosen wisely, it may be that the algorithm does not always converge to the global minimum. In general, though, we may expect that the probability of convergence to a local minimum is decreased by incorporating kernel annealing in the Information Cut algorithm.

Experimentally, we have observed that $\sigma_{start} = 2\sigma$ and $\sigma_{stop} = 0.5\sigma$, with a step size $\Delta_\sigma = (\sigma_{start} - \sigma_{stop})/200$, seem to be a robust choice. In practice, the algorithm should be executed a few times, and then the clustering result corresponding to the lowest Information Cut value should be chosen. We also propose to stop the algorithm when the change in the cost function from one iteration to the next is negligible. For example, one may stop the algorithm when the change in the cost function is less than one percent from one iteration to the next.

In the next experiment, we wish to illustrate the effect of the Parzen window size on the clustering results, still using the data set shown in Figure 1 (a). Recall that by (11), $\sigma = 0.12$ for this data set. Figure 3 shows the error percentage using the Information Cut algorithm for a range of kernel sizes. Here, we have used the clustering result shown in Fig. 1 (b) as the "ground truth." The plot is created by running the algorithm three times for each kernel size, and then picking the best of these trials based on the value of the Information Cut (operating in fixed kernel mode). For $\sigma < 0.05$, the error percentage is very high. In the range $0.11 < \sigma < 0.16$ the error percentage is very low, before it rises again. Note that the kernel size determined by (11) is in the middle of this range. In the range $\sigma > 0.25$, the error percentage is very high again. On average, the algorithm stops after about 25 iterations.



(a) Normalized Cut



(b) Information Cut

Fig. 4. Clustering of data set with different cluster data scales.

4.2 Non-Linear Clusters with Different Data Scales

The annealing procedure comes with a positive side-effect when it comes to coping with clusters of significantly different data scales. In Fig. 4 (a), we show the result obtained by the Normalized Cut algorithm on a data set consisting of three clusters of different data scales. The Normalized Cut algorithm uses a fixed kernel size to determine node affinites, and the clustering result clearly suffers from this property. In fact, the Normalized Cut algorithm did not obtain satisfying results, even when manually tuning the kernel size. The result obtained by the Information Cut algorithm in annealing mode is shown in Fig. 4 (b). The three clusters have all been revealed, even though they have significantly different scales, and are separated by highly non-linear cluster boundaries.

4.3 Pendigits Data Set

This data set was created for pen-based handwritten digit recognition, and is extracted from the UCI repository [33]. 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. We specify $C = 3$ as an input parameter to the algorithm. The clustering results are compared to the known data labels (unknown to the algorithm). The Normalized Cut algorithm obtaines 73.4% correct clustering on this data set. The Information Cut kernel size is automatically determined to be $\sigma = 0.63$. Operating in annealing mode, using $M = 0.2N$ samples for stochastic approximation, the Information Cut algorithm obtaines 84.4% correct clustering. A clear improvement compared to the Normalized Cut method.

One this data set, we investigate more closely the effect of the stochastic sampling approach. The following table shows the Information Cut result (best out of five trials) obtained for a range of M.

M	$0.1N$	$0.2N$	$0.3N$	$0.4N$	$0.5N$
%	83.5	84.4	84.7	85.3	84.1
M	$0.6N$	$0.7N$	$0.8N$	$0.9N$	$1N$
%	84.5	83.8	84.7	85.3	85.0

(21)

We conclude that the stochastic sampling approach is very effective in terms of computational complexity, at virtually no cost in performance.

4.4 Wine Data Set

The wine data set, extracted from the UCI repository, is a well-known benchmark data set. It consists of 178 instances. Each instance has 13 attributes corresponding to chemical properties of three types of wine. The classes consist of 71, 59 and 48 data points, respectively. The normalized cut algorithm performs very well on this data set, obtaining 96.6% correct clustering. The Information Cut algorithm performs consistently equally good or better, obtaining 97.2% correct clustering, operating in annealing mode, with $M = 0.2N$.

4.5 Baseball Player Image

In the following, we perform three image segmentation experiments. The Normalized Cut algorithm was presented as an image segmentation method in [5]. We

perform these experiments to show that the Information Cut algorithm may obtain comparable results also using images.

The (147 × 221) grayscale baseball player image is shown in Fig. 5 (a). It has previously been used to demonstrate the Normalized Cut algorithm in [5]. For each pixel location, we generate three features. One feature is the pixel intensity, and the other two features consist of the pixel location in the two-dimensional plane. There are thus a total of 32487 feature vectors. This is a huge data set. On a Pentium III 1GHz, 512 MBRAM computer, we did not have the capacity even to create the affinity matrix. In fact, in order to be able to execute the Normalized Cut algorithm, we had to randomly select only 1/8 of the pixels to use in the clustering, the rest is classified relative to the clustered feature vectors according to a nearest neighbor rule. To make the algorithms comparable, we used the same subset for the Information Cut algorithm. For the remaining image segmentation experiment, we also ranomly select a subset of the pixels for clustering, and classify the rest accordingly.

Figure 5 (b) show the result obtained by the Information Cut algorithm when partitioning the image into nine segments. Figure 5 (c) show the result obtained by the Normalized Cut algorithm for nine segments. Both methods perform a reasonable segmentation. Some differences are also observed.

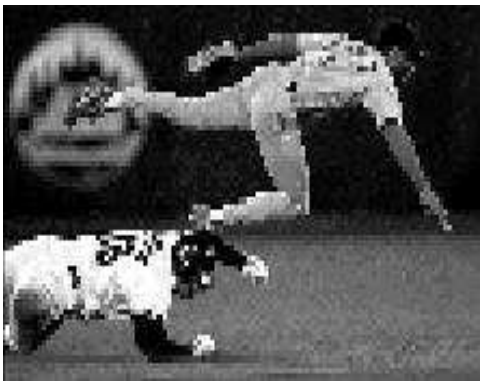
4.6 Northern-Norwegian Boat Image

Figure 6 (a) shows the grayscale version of a (246 × 246) color image of an old traditional-style northern-norwegian boat. In this case, the feature vectors are five-dimensional, i.e. consisting of the rgb-values and the pixel coordinates. Figure 6 (b) and (c) shows a segmentation into nine parts, using the Information Cut and Normalized Cut algorithm, respectively. The results obtained by the two methods differ in this case. Especially when it comes to the ocean surface, it may seem as if the Information Cut algorithm is more sensitive to the inclusion of the pixel coordinates in the feature vectors. The Normalized Cut algorithm produces some very small clusters which are almost invisible in the segmentation.

4.7 Texture Segmentation

Figure 7 (a) shows a (256 × 256) textured image. It consists of five distinct regions, each with specific characteristics. We use the method proposed by Jain and Farrokhnia [34] to produce feature vectors corresponding to pixel locations. Basically, the input image is filtered through a bank of 20 dyadic Gabor filters, producing a 20-dimensional data set. The filtered images will have large energy in regions having a texture characteristic "tuned" to a certain filter.

The result obtained by the Information Cut algorithm is shown in Fig. 7 (b). The



(a) Original (147 × 221)

(b) Information Cut, 9 segments

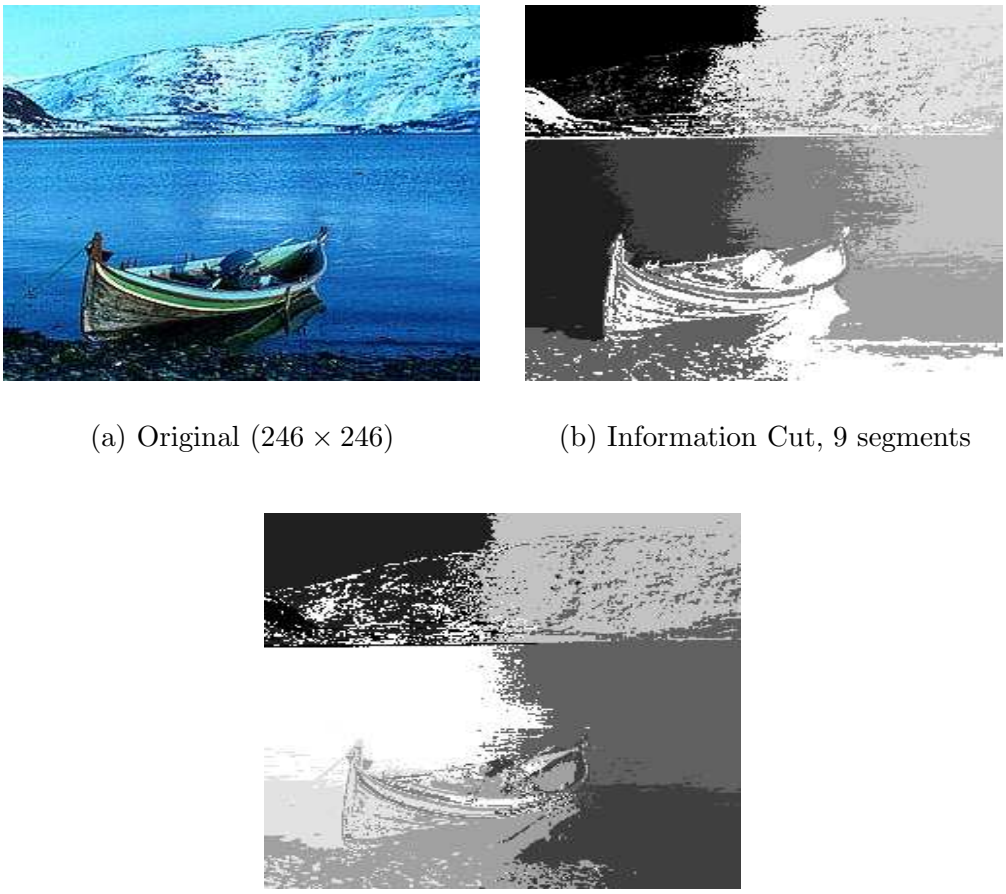
(c) Normalized Cut, 9 segments

Fig. 5. Segmentation of the *baseball player* image.

five segments clearly correspond to the different textured regions in the input image, with some inaccuracies on the texture boundaries (texture boundaries indicated by the white lines). The Information Cut misclassifies 3.0% of the pixels. The result obtained by the Normalized Cut algorithm is shown in Fig. 7 (c). It also obtaines a reasonable result, but not as good as the Information Cut. It misclassifies 5.1% of the pixels.

5 Conclusions

We have derived an interesting connection between a particular information theoretic divergence measure, namely the Cauchy-Schwarz divergence, and the graph theoretic *cut*. The key component in this derivation is the Parzen window technique for probability density estimation. The new graph theoretic cost function, named the Information Cut, provides a theoretically well-founded normalization to the traditional *cut*-cost. This connection between information theory and graph theory can



(a) Original (246 × 246)

(b) Information Cut, 9 segments

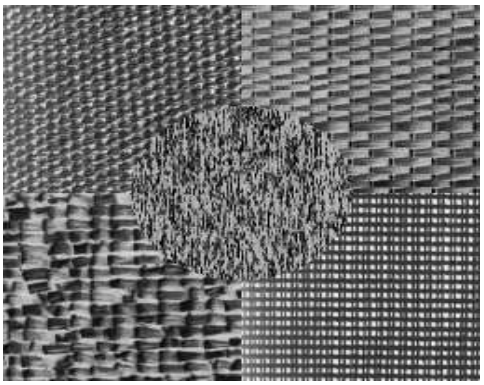
(c) Normalized Cut, 9 segments

Fig. 6. Segmentation of old traditional-style northern-norwegian boat image.

not be obtained for other divergence measures, such as the Kullback-Leibler, or the Chernoff divergences.

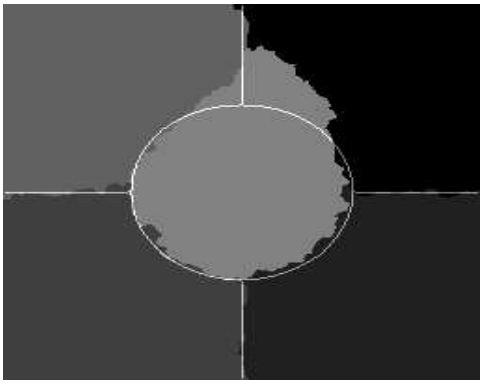
A new algorithm for clustering and image segmentation based on minimizing the Information Cut with respect to cluster membership variables has been derived. It has been shown that the resulting algorithm may perform comparable or better than the state-of-the art graph-based method, namely the Normalized Cut technique. Moreover, we have argued that our optimization approach has benefits with respect to computational complexity and memory requirements. We have shown that our strategy for dealing with the problem of convergence to a local minimum, namely kernel annealing, may also have the positive side-effect that we are better able to handle different cluster data scales.

Importantly, in this paper, two domains of research has been coupled, i.e. graph-based clustering and non-parametric density estimation in terms of Parzen windowing. Consequently, it was observed that the Parzen window width directly determines the affinity measure used to calculate edge-weights in the graph. It is well-known that it is crucial for any graph-based method to determine this pa-



(a) Original (256×256)

(b) Information Cut, 5 segments



(c) Normalized Cut, 5 segments

Fig. 7. Segmentation of a textured image consisting of five distinct regions.

rameter appropriately, but few data-driven methods exist. We used the simplest approach for data-driven kernel size selection, namely Silverman's rule. However, more advanced techniques can easily be incorporated. Hence, we may study the statistics literature in non-parametric density estimation, in order to gain new tools for computing edge-weights in the graph. In future work we will investigate whether local and anisotropic Parzen windows used to calculate graph affinities may further improve the clustering and image segmentation results. Another possibility is to use the k nearest neighbors density estimation technique. We will also investigate the use of the fast Gauss transform [35] to speed up the Parzen window density estimation, and hence speed up the clustering algorithm. Another issue which should be studied is the annealing scheme. It may be possible to connect the annealing parameter to the local curvature of the cost function at each iteration step, thus making the annealing scheme user independent.

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.

Appendix

A.

Eq. (3) can be rewritten as

$$D_{CS} = -\log \frac{E_{p_1}\{p_2(\mathbf{x})\}}{\sqrt{E_{p_1}\{p_1(\mathbf{x})\}E_{p_2}\{p_2(\mathbf{x})\}}}, \tag{22}$$

where $E_p\{\cdot\}$ denotes the expectation operator with respect to the density p. Using the sample mean to estimate the expectations, we obtain $E_{p_1}\{p_2(\mathbf{x})\} \approx \frac{1}{N_1}\sum_{i=1}^{N_1} p_2(\mathbf{x}_i)$ $= \frac{1}{N_1}\sum_{i=1}^{N_1}\frac{1}{N_2}\sum_{j=1}^{N_2} W(\mathbf{x}_i,\mathbf{x}_j) = \frac{1}{N_1 N_2}\sum_{i,j=1}^{N_1,N_2} W(\mathbf{x}_i,\mathbf{x}_j)$, where W is some (non-Gaussian) Parzen window. In a similar manner, $E_{p_1}\{p_1(\mathbf{x})\} \approx \frac{1}{N_1 N_1}\sum_{i,i'=1}^{N_1,N_1} W(\mathbf{x}_i,\mathbf{x}_{i'})$ and $E_{p_2}\{p_2(\mathbf{x})\} \approx \frac{1}{N_2 N_2}\sum_{j,j'=1}^{N_2,N_2} W(\mathbf{x}_j,\mathbf{x}_{j'})$, such that we obtain

$$IC(\mathcal{G}_1,\mathcal{G}_2) = \frac{\sum_{i,j=1}^{N_1,N_2} k_{ij}}{\sqrt{\sum_{i,i'=1}^{N_1,N_1} k_{ii'}\sum_{j,j'=1}^{N_2,N_2} k_{jj'}}}, \tag{23}$$

where we have defined $W(\mathbf{x}_l,\mathbf{x}_{l'}) = k_{ll'}$.

B.

Let $IC = \frac{U}{V}$, where $U = \frac{1}{2}\sum_{i,j=1}^{N,N}\left(1-\mathbf{m}_i^T\mathbf{m}_j\right)k_{ij}$, $V = \sqrt{\prod_{c=1}^{C} v_c}$ *and* $v_c = \sum_{i,j=1}^{N,N} m_{ic}m_{jc}k_{ij}$. Hence

$$\frac{\partial IC}{\partial \mathbf{m}_i} = \frac{V\frac{\partial U}{\partial \mathbf{m}_i} - U\frac{\partial V}{\partial \mathbf{m}_i}}{V^2} \tag{24}$$

$$\frac{\partial U}{\partial \mathbf{m}_i} = -\sum_{j=1}^{N}\mathbf{m}_j k_{ij}, \tag{25}$$

$$\frac{\partial V}{\partial \mathbf{m}_i} = \frac{1}{2} \sum_{c'=1}^{C} \sqrt{\frac{\prod_{c \neq c', c=1}^{C} v_c}{v_{c'}}} \frac{\partial v_{c'}}{\partial \mathbf{m}_i}, \tag{26}$$

where $\frac{\partial v_{c'}}{\partial \mathbf{m}_i} = \begin{bmatrix} 0 & \dots & 2\sum_{j=1}^{N} m_{jc'} k_{ij} & \dots & 0 \end{bmatrix}^T$. Thus, only element number c' of this vector is nonzero.

ROBERT JENSSEN'S research interests are in non-parametric information theoretic learning, kernel methods, spectral clustering and independent component analysis. Jenssen has received "Honorable Mention" for the 2003 Pattern Recognition Socitety Best Paper Award, and received the ICASSP 2005 Outstanding Student Paper Award. He serves on the program committee of the IEEE MLSP conference.

DENIZ ERDOGMUS' research interests are in adaptive non-linear, and statistical signal processing. Erdogmus has received the International Neural Network Society 2004 Young Investigator Award and the IEEE Signal Processing Society 2003 Young Author Best Paper Award. He has served on the program committee of numerous conferences.

KENNETH E. HILD II'S research interests are in independent component analysis and other advanced signal and image processing techniques, with special application emphasis on magnetoencephalographic data. He serves on the program committee of the IEEE MLSP conference.

JOSE C. PRINCIPE'S research interests are in computational neuro-engineering, adaptive systems theory, machine learning and nonlinear dynamics. He is a Bell-South Professor, a fellow of the IEEE, Editor-in-Chief of the IEEE Trans. Biomedical Engineering and the President of the INNS. He has rececived several awards, and has served on numerous program committees.

TORBJØRN ELTOFT'S research interests are in remote sensing, image and signal analysis and artificial neural networks . He received the year 2000 Outstanding Paper Award from the IEEE Neural Networks Society, and "Honorable Mention" for the 2003 Pattern Recognition Socitety Best Paper Award.

References

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

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

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

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

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

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

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

[20] 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.

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

[22] 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.

[23] 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.

[24] R. Jenssen, D. Erdogmus, K. E. Hild, J. C. Principe, and T. Eltoft. Optimizing the cauchy-schwarz pdf distance for information theoretic, non-parametric clustering. In *Proceedings of International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, page (to appear), St. Augustine, USA, November 9-11, 2005.

[25] 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.

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

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

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

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

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

[31] 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.

[32] 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.

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

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

[35] A. Elgammal, R. Duraiswami, and L. Davis. The Fast Gauss Transform for Efficient Kernel Density Evaluation with Applications in Computer Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1499–1504, 2003.