Optimizing the Cauchy-Schwarz PDF Distance for Information Theoretic, Non-Parametric Clustering *

Robert Jenssen **[1], Deniz Erdogmus[2], Kenneth E. Hild II[3], Jose C. Principe[4], and Torbjørn Eltoft[1]

[1] Department of Physics
University of Tromsø, N - 9037 Tromsø, Norway

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

[3] Department of Radiology
University of California, San Francisco, CA. 94143, USA

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

Abstract. This paper addresses the problem of efficient information theoretic, non-parametric data clustering. We develop a procedure for adapting the cluster memberships of the data patterns, in order to maximize the recent Cauchy-Schwarz (CS) probability density function (pdf) distance measure. Each pdf corresponds to a cluster. The CS distance is estimated analytically and non-parametrically by means of the Parzen window technique for density estimation. The resulting form of the cost function makes it possible to develop an efficient adaption procedure based on constrained gradient descent, using stochastic approximation of the gradients. The computational complexity of the algorithm is $O(MN)$, $M \ll N$, where N is the total number of data patterns and M is the number of data patterns used in the stochastic approximation. We show that the new algorithm is capable of performing well on several odd-shaped and irregular data sets.

1 Introduction

In data analysis, it is often desirable to partition, or cluster, a data set into subsets, such that members within subsets are more similar to each other according to some criterion, than to members of other subsets. Clustering has many important applications in computer vision and pattern recognition. See for example ref. [1] for a review.

* This work was partially supported by NSF grants ECS-9900394 and EIA-0135946
** robertj@phys.uit.no Phone: (+47) 776 46493 Fax: (+47) 776 45580

Most of the traditional algorithms, such as fuzzy K-means [2] and the expectation-maximization algorithm for a Gaussian mixture model (EMGMM) [3], work well for hyper-spherical and hyper-elliptical clusters, since they are often optimized based on a second order statistics criterion. Therefore, in recent years, the main thrust in clustering has been towards developing efficient algorithms capable of handling odd-shaped and highly irregular clusters.

Information theoretic methods appear as particularly appealing alternatives as clustering cost functions when it comes to capturing all the structure in a data set. The reason is that pdf distance measures in theory do capture all the information contained in the data distributions in question. Several information theoretic approaches to clustering have been proposed in recent years, see for example refs. [4–7]. The problem with many such methods is often that the information theoretic measure can be difficult to estimate. Analytical estimation most often requires the user to choose a parametric model for the data distributions. Hence, the clustering algorithm will only perform well if the parametric model matches the actual densities. Also, the optimization of the cost function is often computationally demanding.

In this paper, we address the problem of efficient information theoretic, non-parametric data clustering. We develop a procedure for adjusting the cluster memberships of the data points, which seeks to maximize the CS pdf distance measure. Since the estimated pdfs, at each iteration cycle, are based on the current clusters, the approach is to assign the memberships of the data such that the CS distance between the obtained clusters is maximized. The CS distance can be estimated analytically and non-parametrically by means of the Parzen window technique for density estimation. Hence, the cost function captures all the statistical information contained in the data.

By estimating the cluster pdfs using the Parzen window technique, the CS distance can be expressed in terms of cluster memberships with respect to a predetermined number of clusters. Of course, in clustering, the memberships are not known beforehand, and have to be initialized randomly. The adaption procedure for these memberships, maximizing the CS cost function, is carried out by means of the Lagrange multiplier formalism. The procedure can be considered a constrained gradient descent search, with built in variable step-sizes for each coordinate direction.

The resulting algorithm has a complexity of order $O(N^2)$, where N is the number of data patterns. In practical clustering problems, the data sets may be very large. Thus, it is of crucial importance to reduce the complexity of the algorithm. To achieve this goal, we derive a stochastic approximation approach to estimating the gradients used in the clustering rule. Instead of calculating the gradients based on information from the memberships corresponding to all the data points, we stochastically sample the membership space, using only $M \ll N$ randomly selected membership functions and their corresponding data points, to calculate the gradients. As a result, we obtain an efficient information theoretic clustering algorithm of only $O(MN)$ complexity.

The Parzen window size will also be used to avoid a pitfall of gradient descent learning in non-convex cost functions, i.e., the convergence to a local optimum of the cost function. We show that in our algorithm, this problem can to a high degree be avoided, by allowing the size of the Parzen kernel to be annealed over a range of values around the optimally estimated value. The effect of using a large kernel compared to the optimal kernel size, is to obtain an over-smoothed version of the CS cost function, such that many local optima are eliminated. As the algorithm converges toward the optimum of the smoothed CS distance, the kernel size is continuously decreased, leading the algorithm toward the true global optimum. We propose a method to select a suitable annealing scheme based on the optimal Parzen kernel, which is, however, rather heuristic at this point.

The organization of this paper is as follows. In section 2, we review the Cauchy-Schwarz pdf distance measure. In section 3, we develop the Lagrange multiplier optimization procedure, and show how the gradients can be stochastically approximated to obtain an efficient clustering algorithm. We present some clustering experiments in section 4, and make our concluding remarks in section 5.

2 Cauchy-Schwarz PDF Distance

Based on the Cauchy-Schwarz inequality; $||\mathbf{x}||^2 \; ||\mathbf{y}||^2 \geq (\mathbf{x}^T\mathbf{y})^2$, the following holds;

$$-\log \frac{\mathbf{x}^T\mathbf{y}}{\sqrt{||\mathbf{x}||^2 \; ||\mathbf{y}||^2}} \geq 0. \tag{1}$$

By replacing inner products between vectors in (1), by inner products between pdfs, i.e. $\langle p, q \rangle = \int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}$, we define the following distance measure [8]

$$D(p,q) = -\log \frac{\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}}{\sqrt{\int p^2(\mathbf{x})d\mathbf{x} \; \int q^2(\mathbf{x})d\mathbf{x}}} \geq 0. \tag{2}$$

We refer to $D(p,q)$ as the Cauchy-Schwarz pdf distance measure. It can be seen that $D(p,q)$ is always non-negative, it obeys the identity property, and it is also symmetric. The $D(p,q)$ goes to infinity when the overlap between the two pdfs goes to zero. The measure does however not obey the triangle inequality, such that it does not satisfy the strictly mathematical definition of a distance measure.

Since the logarithm is a monotonic function, maximization of $D(p,q)$ is equivalent to minimization of the argument of the log in (2). In this paper, we refer to this quantity as $J(p,q)$, and the goal is to develop an efficient minimization scheme for this quantity.

Assume that we estimate $p(\mathbf{x})$ based on the data points in cluster $C_1 = \{\mathbf{x}_i\}, \; i = 1, \ldots, N_1$, and $q(\mathbf{x})$ based on $C_2 = \{\mathbf{x}_j\}, \; j = 1, \ldots, N_2$. By the Parzen

[9] method

$$\hat{p}(\mathbf{x}) = \frac{1}{N_1}\sum_{i=1}^{N_1} G(\mathbf{x}-\mathbf{x}_i, \sigma^2\mathbf{I}),$$
$$\hat{q}(\mathbf{x}) = \frac{1}{N_2}\sum_{j=1}^{N_2} G(\mathbf{x}-\mathbf{x}_j, \sigma^2\mathbf{I}), \tag{3}$$

where we have used symmetric Gaussian kernels, $G(\mathbf{x}, \mathbf{\Sigma})$, where $\mathbf{\Sigma} = \sigma^2\mathbf{I}$. According to the convolution theorem for Gaussians, the following relation holds

$$\int G(\mathbf{x}-\mathbf{x}_i, \sigma^2\mathbf{I}) G(\mathbf{x}-\mathbf{x}_j, \sigma^2\mathbf{I}) d\mathbf{x} = G_{ij,2\sigma^2\mathbf{I}}, \tag{4}$$

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

Thus, when we plug the Parzen pdf estimates of (3) into (2), and utilize (4), we obtain

$$\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x} \approx \frac{1}{N_1 N_2}\sum_{i,j=1}^{N_1,N_2} G_{ij,2\sigma^2\mathbf{I}}, \tag{5}$$

$$\int p^2(\mathbf{x})d\mathbf{x} \approx \frac{1}{N_1^2}\sum_{i,i'=1}^{N_1,N_1} G_{ii',2\sigma^2\mathbf{I}}, \tag{6}$$

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

$$J(p,q) = \frac{\frac{1}{N_1 N_2}\sum_{i,j=1}^{N_1,N_2} G_{ij,2\sigma^2\mathbf{I}}}{\sqrt{\frac{1}{N_1^2}\sum_{i,i'=1}^{N_1,N_1} G_{ii',2\sigma^2\mathbf{I}} \; \frac{1}{N_2^2}\sum_{j,j'=1}^{N_2,N_2} G_{jj',2\sigma^2\mathbf{I}}}}. \tag{7}$$

For each data pattern $\mathbf{x}_i,\ i = 1,\ldots,N,\ N = N_1 + N_2$, we now define a membership vector $\mathbf{m}_i$. If $\mathbf{x}_i$ belongs to cluster C_1 (C_2), the corresponding crisp membership vector equals $\mathbf{m}_i = [1,0]^T$ ($[0,1]^T$). This allows us to rewrite (7) as a function of the memberships, obtaining;

$$J(p,q) = \frac{\frac{1}{2}\sum_{i,j=1}^{N,N}\left(1 - \mathbf{m}_i^T\mathbf{m}_j\right) G_{ij,2\sigma^2\mathbf{I}}}{\sqrt{\prod_{k=1}^{2}\sum_{i,j=1}^{N,N} m_{ik}m_{jk}G_{ij,2\sigma^2\mathbf{I}}}}, \tag{8}$$

where m_{ik} (m_{jk}), $k = 1,2$, denotes element number k of $\mathbf{m}_i$ ($\mathbf{m}_j$). In the sequel we will make explicit that the variable quantities in (8) are the membership vectors, thus, we will use the notation $J(\mathbf{m}_1,\ldots,\mathbf{m}_N)$ instead of $J(p,q)$.

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

$$J(\mathbf{m}_1,\ldots,\mathbf{m}_N) = \frac{\frac{1}{2}\sum_{i,j=1}^{N,N}\left(1 - \mathbf{m}_i^T\mathbf{m}_j\right) G_{ij,2\sigma^2\mathbf{I}}}{\sqrt{\prod_{k=1}^{K}\sum_{i,j=1}^{N,N} m_{ik}m_{jk}G_{ij,2\sigma^2\mathbf{I}}}}, \tag{9}$$

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

The cost function $J(\mathbf{m}_1, \ldots, \mathbf{m}_N)$ is related to the cluster evaluation function used by Gokcay and Principe [10]. They basically clustered based on the numerator of (9), which can be considered equivalent to a "between-cluster" Renyi entropy measure. Their clustering technique was based on calculating the cluster evaluation function for all clustering possibilities, hence impractical for anything but very small data sets. We incorporate the "within-cluster" Renyi entropies in the cost function, which are equivalent to the quantities in the denominator of (9). This helps balance the cost function, and avoids problems such as obtaining a minimum of the cost function when only one data point is isolated in a cluster, and all the other data points in the remaining cluster. In addition, in the following we will derive an efficient optimization technique for minimizing $J(\mathbf{m}_1, \ldots, \mathbf{m}_N)$.

We assume a-priori knowledge about the number, K, of clusters inherit in the data set. This may seem to be a strict assumption, and in some cases it probably is. However, much research has been conducted with regard to estimating the number of clusters present in a data set. See e.g. [11] for an overview of different cluster indicies.

3 Lagrange Optimization

In order to minimize (9) using differential calculus techniques, we need to fuzzify the membership vectors such that $\mathbf{m}_i \in [0, 1],\ i = 1, \ldots, N$. Accordingly, we suggest to solve the following constrained optimization problem

$$\min_{\mathbf{m}_1, \ldots, \mathbf{m}_N} \ J(\mathbf{m}_1, \ldots, \mathbf{m}_N), \tag{10}$$

subject to $\mathbf{m}_j^T \mathbf{1} - 1 = 0, \quad j = 1, \ldots, N$, where $\mathbf{1}$ is a K-dimensional ones-vector. Now we make a convenient change of variables. Let $m_{ik} = v_{ik}^2,\ k = 1, \ldots, K$. Consider

$$\min_{\mathbf{v}_1, \ldots, \mathbf{v}_N} \ J(\mathbf{v}_1, \ldots, \mathbf{v}_N), \tag{11}$$

subject to $\mathbf{v}_j^T \mathbf{v}_j - 1 = 0, \quad j = 1, \ldots, N$. The constraints for the problem stated in (11) are equivalent to the constraints for (10).The optimization problem, (11), amounts to adjusting the vectors $\mathbf{v}_i,\ i = 1, \ldots, N$, such that

$$\frac{\partial J}{\partial \mathbf{v}_i} = \left(\frac{\partial J}{\partial \mathbf{m}_i}^T \frac{\partial \mathbf{m}_i}{\partial \mathbf{v}_i} \right)^T = \mathbf{\Gamma} \frac{\partial J}{\partial \mathbf{m}_i} \rightarrow \mathbf{0}, \tag{12}$$

where $\mathbf{\Gamma} = \mathrm{diag}(2\sqrt{m_{i1}}, \ldots, 2\sqrt{m_{iK}})$. We force all elements $2\sqrt{m_{ik}},\ k = 1, \ldots, K$, to always be positive by adding a small positive constant ϵ during each membership update. Hence, $\frac{\partial J}{\partial \mathbf{v}_i} \rightarrow 0$ implies $\frac{\partial J}{\partial \mathbf{m}_i} \rightarrow 0$. Thus, these scalars can be interpreted as variable step-sizes built into the gradient descent search process,

as a consequence of the change of variables that we made. We will return to the derivation of $\frac{\partial J}{\partial \mathbf{m}_i}$ in subsection 3.2, and to the stochastic approximation of this quantity.

The necessary conditions for the solution of (11) are commonly generated by constructing the Lagrange function, given by

$$L = J(\mathbf{v}_1, \ldots, \mathbf{v}_N) + \sum_{j=1}^{N} \lambda_j \left(\mathbf{v}_j^T \mathbf{v}_j - 1 \right), \tag{13}$$

where $\lambda_j,\ j = 1, \ldots, N$, are the *Lagrange multipliers*. The necessary conditions for the extremum of L are given by

$$\frac{\partial L}{\partial \mathbf{v}_i} = \frac{\partial J}{\partial \mathbf{v}_i} + \sum_{k=1}^{N} \lambda_k \frac{\partial}{\partial \mathbf{v}_i} \left(\mathbf{v}_k^T \mathbf{v}_k - 1 \right) = \mathbf{0}, \tag{14}$$

$$\frac{\partial L}{\partial \lambda_j} = \mathbf{v}_j^T \mathbf{v}_j - 1 = 0, \tag{15}$$

for $i, j = 1, \ldots, N$. From (14) we derive the following *fixed-point adaption rule* for the vector $\mathbf{v}_i$ as follows

$$\frac{\partial J}{\partial \mathbf{v}_i} + 2\lambda_i \mathbf{v}_i = \mathbf{0} \Rightarrow \mathbf{v}_i^+ = -\frac{1}{2\lambda_i} \frac{\partial J}{\partial \mathbf{v}_i}, \tag{16}$$

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

We solve for the Lagrange multipliers, $\lambda_i,\ i = 1, \ldots, N$, by evaluating (15), yielding

$$\lambda_i = \frac{1}{2} \sqrt{\frac{\partial J}{\partial \mathbf{v}_i}^T \frac{\partial J}{\partial \mathbf{v}_i}}. \tag{17}$$

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

We initialize the membership vectors randomly according to a uniform distribution. That way $\mathbf{m}_i \in [0, 1]\ \forall i$, even though the constraint of (10) is not obeyed. We have observed that after the first iteration through the algorithm, the constraint is always obeyed. Better initialization schemes may be used, although in our experiments, the algorithm is very little affected by the actual initialization used.

3.1 Kernel size and annealing scheme

In section 2, the same kernel size, σ, was used in the Parzen estimate of both (all) the pdfs of the clusters. Obviously, to obtain a perfect pdf estimate for each cluster, this assumption may not be valid. But since we don't know which data points belong to which cluster (since this is exactly what we are trying to determine) it is impossible to obtain a separate kernel size for each cluster.

Given an input data set, the best we can do is to estimate the optimal kernel size σ based on the whole data set. In section 4, we show that for the purpose of clustering, using a single kernel size for each cluster gives promising results, even though the underlying densities are not necessarily optimally estimated.

We will use Silverman's rule-of-thumb to determine the optimal kernel size with respect to a mean integrated square error criterion between the estimated and the actual pdf. It is given by [12]

$$\sigma_{\text{opt}} = \sigma_X \left\{ 4N^{-1}(2d+1)^{-1} \right\}^{\frac{1}{d+4}}, \tag{18}$$

where d is the dimensionality of the data and $\sigma_X^2 = d^{-1} \sum_i \Sigma_{X_{ii}}$ and $\Sigma_{X_{ii}}$ are the diagonal elements of the sample covariance matrix.

The new CS-clustering algorithm that we propose can be operated in a fully automatic mode by selecting the kernel size using (18), assuming that the correct number of clusters, K, has been estimated beforehand. Hence no user-specified parameters are needed. However, since the CS-cost function is non-convex, it may exhibit more than one optimum. For many data sets, the algorithm may always converge to the correct solution, but for other data sets, it may in some cases converge to a local non-optimal solution.

The Parzen windowing makes it possible to incorporate a learning strategy into the algorithm to help avoid local minima. The kernel size is allowed to be annealed over a range of values around the optimal value. We start out with a relatively large kernel size, which has the effect of smoothing out local minima of the cost function. As the algorithm converges toward the global minimum of the smoothed cost function, which is biased wrt. the location of the true minimum, the kernel size is continuously annealed, such that the minimum of the smoothed cost function gets more and more aligned with the true minimum. By incorporating the annealing into the algorithm, we can be more certain that the solution obtained is close to the desired solution.

3.2 Stochastic approximation

In this subsection we examine the stochastic approximation approach for calculating the gradient $\frac{\partial J}{\partial \mathbf{m}_i}$.

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

$$\begin{aligned} U &= \frac{1}{2} \sum_{i,j=1}^{N,N} \left(1 - \mathbf{m}_i^T \mathbf{m}_j\right) G_{ij,2\sigma^2\mathbf{I}}, \\ V &= \sqrt{\prod_{k=1}^{K} v_k} \text{ and } v_k = \sum_{i,j=1}^{N,N} m_{ik} m_{jk} G_{ij,2\sigma^2\mathbf{I}}. \end{aligned} \tag{19}$$

Hence

$$\frac{\partial J}{\partial \mathbf{m}_i} = \frac{V \frac{\partial U}{\partial \mathbf{m}_i} - U \frac{\partial V}{\partial \mathbf{m}_i}}{V^2}, \tag{20}$$

$$\frac{\partial U}{\partial \mathbf{m}_i} = -\sum_{j=1}^{N} \mathbf{m}_j G_{ij,2\sigma^2\mathbf{I}}, \tag{21}$$

$$\frac{\partial V}{\partial \mathbf{m}_i} = \frac{1}{2}\sum_{k'=1}^{K} \sqrt{\frac{\prod_{\substack{k=1 \\ k\neq k'}}^{K} v_k}{v_{k'}}} \frac{\partial v_{k'}}{\partial \mathbf{m}_i}, \tag{22}$$

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

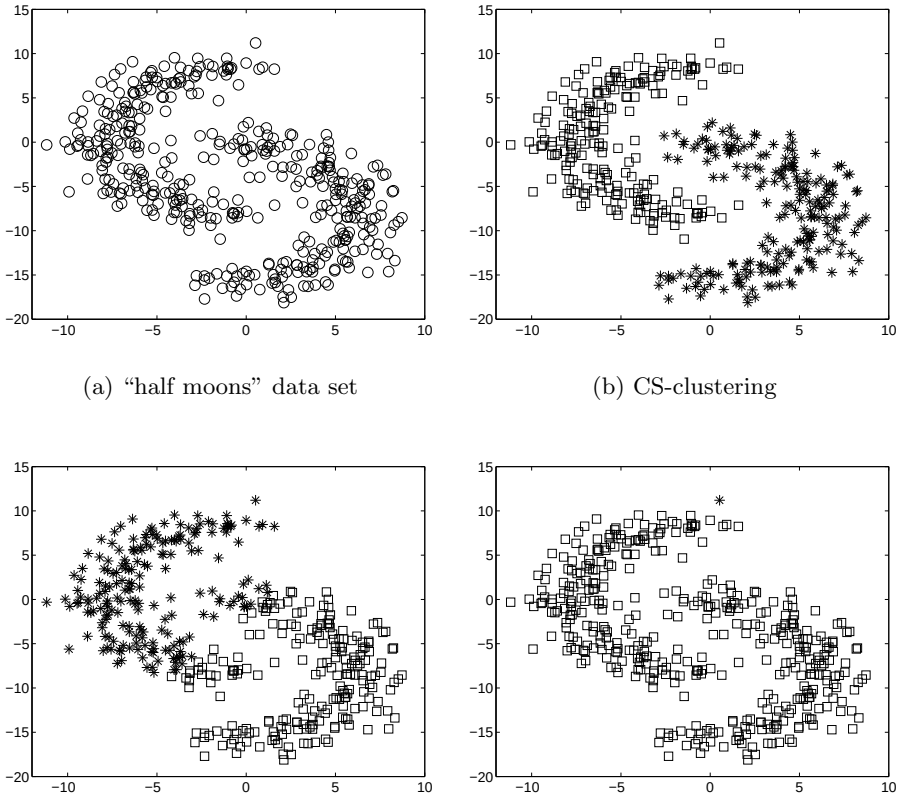
The key point to note, is that we can calculate all quantities of interest in (20), by determining (21), for $\forall i$. Since (21) is a sum over N elements, calculating all these quantities is an $O(N^2)$ procedure. An $O(N^2)$ algorithm may become intractable for large data sets. To reduce complexity, we estimate (21) by *stochastically sampling* the membership space, and utilize M randomly selected membership vectors, and corresponding data points, to compute

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

as an approximation to (21). 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 selecting M to be as small as 15% of N.

4 Clustering Experiments

In this section we report clustering results on two artificially created data sets, and one real. In all experiments, we use (18) to estimate the kernel size with respect to Parzen pdf estimation. The upper limit of the kernel size, which we start out with in the annealing procedure, is chosen to be $\sigma_{\text{upper}} = 2\sigma_{\text{opt}}$, and the lower limit is selected as $\sigma_{\text{lower}} = 0.5\sigma_{\text{opt}}$. The kernel size is linearly decreased using a step size $\Delta_\sigma = (\sigma_{\text{upper}} - \sigma_{\text{lower}})/100$. If convergence is not obtained when reaching σ_{lower}, the algorithm continues using σ_{lower} as the kernel size. These values are selected based on our experimental experience. It should be said that the algorithm is quite robust with regard to these values. Also, the value of M is always selected as 15% of the value of N (rounded to the nearest integer). Our experiments show that even thought we only use a few randomly chosen points to estimate the gradients, the results are as good as utilizing the whole data set. The memberships are initialized as proposed in section 3, and the constant $\epsilon = 0.05$. In order to stop the algorithm, we examine the crisp memberships every tenth iteration. If there is no change in crisp memberships over these ten iterations, it is assumed that the algorithm has either converged to a reasonable solution, or that the algorithm is trapped in a local minimum from which it cannot escape. Hence, when the algorithm terminates, it has in practice converged at least ten iterations earlier.



(a) "half moons" data set

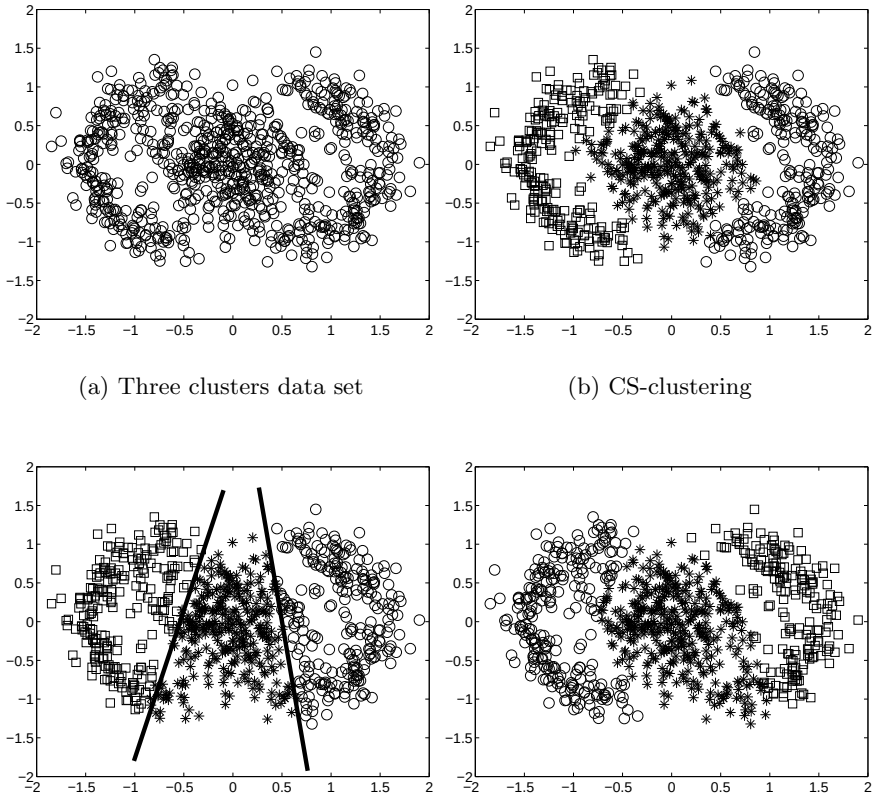
(b) CS-clustering

(c) EMGMM-clustering

(d) SL-clustering

Fig. 1. The CS-clustering algorithm performs very well on this two-cluster data set, where the boundary between the clusters is highly non-linear, and there are some overlap.

In our first experiment, we consider the data set shown in Fig. 1 (a). A human can observe that it contains two "half-moon"-shaped clusters with a highly non-linear cluster boundary. There are totally $N = 419$ data patterns. The data set is clustered 20 times using the CS-clustering algorithm. In absolutely all trials, a result similar to that shown in Fig. 1 (b) is produced, after on average about 100 iterations. It can be seen that the clustering reveals the structure of the data set. It should be said that a similar result is also obtained in 80% of the trials using the fixed kernel mode, that is, the kernel is not annealed. Hence, in fixed kernel mode, the algorithm converges to a local optimum in 20% of the trials. For comparison, a typical result using the EMGMM algorithm is shown in Fig. 1 (c). The EMGMM algorithm never obtaines the desired result, but always produces a near-linear cluster boundary. The same is the case for the



(a) Three clusters data set

(b) CS-clustering

(c) FCM-clustering

(d) Normalized Cut clustering

Fig. 2. Data set consisting of three clusters used in second clustering experiment.

Fuzzy K-means algorithm, which produces a result similar to the EMGMM (not shown). Also, the result using the single-link clustering algorithm [1] is shown in Fig. 1 (d). It can be seen that it isolates a single point in one cluster, and links together all the rest. This behavior is typical for the single-link algorithm when the clusters have some overlap. Fortunately, the CS-clustering algorithm shows no such tendency. Note that all the clustering methods we compare with are popular and often used in practice.

In the second experiment, we cluster the data set shown in Fig. 2 (a). It contains $N = 819$ data patterns. As can be observed, there seems to be three clusters, but the boundaries are not very clear. Consistently, the CS-algorithm produces a clustering result as shown in Fig. 2 (b), after on average about 120 iterations. The result clearly seems to be reasonable, considering the structure of the data set. For comparison, the result obtained using fuzzy K-means is

shown in Fig. 2 (c). The linear cluster boundaries this method produces are shown by the straight lines, obviously not capturing the non-linear nature of the data. The result obtained using the EMGMM algorithm is quite similar, and is not shown. The single-link algorithm fails completely on this kind of data, because the data is noisy. We also include a comparison to a recent graph-based clustering algorithm known as the Normalized Cut method [13]. The scale parameter used in this method to define graph edge-weights was recommended by the authors to be in the range $10 - 20\%$ of the total range of the Euclidean feature vector distances. We use 15%. The resulting clustering is shown in Fig. 2 (d). It is clearly an improvement over fuzzy K-means, but seems not to capture the cluster structure to the same degree as our proposed method.

As a final experiment, the Wisconsin breast-cancer (WBC) data set [14] is clustered. It consists of 683 data points (444 benign and 239 malignant). WBC is a nine-dimensional dataset with features related to clump thickness, uniformity of cell size, shape, and so forth. See [14] for details. On average, we obtained a correct classification rate of 94.5%, which is comparable to the best results reported for other clustering schemes on this data set.

5 Conclusions

In this paper, we have developed a clustering algorithm that is based on optimizing the Cauchy-Schwarz information theoretic distance measure between densities. The optimization is carried out using the Lagrange multiplier formalism, and can be considered a constrained gradient descent search. The gradients are stochastically approximated, reducing the complexity from $O(N^2)$ to $O(MN)$, $M \ll N$. We have shown that the algorithm performs well on several data sets, and that it is capable of clustering data sets where the cluster boundaries are highly non-linear. We attribute this property to the information theoretic metric we use, combined with non-parametric Parzen density estimation.

Jenssen et al. [15] in fact discovered a relationship between the Cauchy-Schwarz pdf distance and the graph theoretic *cut*. This means that our proposed method can also be considered to belong to the family of graph-based clustering cost functions, and it is hence related to the Normalized Cut method and spectral clustering. However, in our method, there is no need to compute eigenvectors, which is known to be a computationally demanding procedure. In future work, we will further pursue this link between our information theoretic approach and graph theory. See also [16] for comments on this link.

References

1. A. K. Jain, M. N. Murty, and P. J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.
2. J. C. Bezdek, "A Convergence Theorem for the Fuzzy Isodata Clustering Algorithms," *IEEE Transactions on Pattern Analysis and Machine Learning*, vol. 2, no. 1, pp. 1–8, 1980.

3. G. J. McLachlan and D. Peel, *Finite Mixture Models*, John Wiley & Sons, New York, 2000.
4. K. Rose, E. Gurewitz, and G. C. Fox, "Vector Quantization by Deterministic Annealing," *IEEE Transactions on Information Theory*, vol. 38, no. 4, pp. 1249–1257, 1992.
5. T. Hofmann and J. M. Buhmann, "Pairwise Data Clustering by Deterministic Annealing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, pp. 1–14, 1997.
6. S. J. Roberts, R. Everson, and I. Rezek, "Maximum Certainty Data Partitioning," *Pattern Recognition*, vol. 33, pp. 833–839, 2000.
7. N. Tishby and N. Slonim, "Data Clustering by Markovian Relaxation and the Information Bottleneck Method," in *Advances in Neural Information Processing Systems, 13*, MIT Press, Cambridge, 2001, pp. 640–646.
8. 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.
9. E. Parzen, "On the Estimation of a Probability Density Function and the Mode," *The Annals of Mathematical Statistics*, vol. 32, pp. 1065–1076, 1962.
10. E. Gokcay and J. Principe, "Information Theoretic Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 158–170, 2002.
11. G. W. Milligan and M. C. Cooper, "An Examination of Procedures for Determining the Number of Clusters in a Data Set," *Phychometrica*, pp. 159–179, 1985.
12. B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London, 1986.
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. O.L Mangasarian and W. H. Wolberg, "Cancer Diagnosis via Linear Programming," *SIAM News*, vol. 5, pp. 1–18, 1990.
15. 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*, Toulouse, France, September 17-19, 2003, pp. 459–468.
16. 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.