Information Force Clustering using Directed Trees *

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

[1] Computational NeuroEngineering Laboratory
Department of Electrical and Computer Engineering
University of Florida
Gainesville FL. 32611, USA

[2] Electrical Engineering Group
Department of Physics
University of Tromsø
N - 9037 Tromsø, Norway

Abstract. We regard a data pattern as a physical particle experiencing a force acting on it imposed by an overall "potential energy" of the data set, obtained via a non-parametric estimate of Renyi's entropy. The "potential energy" is called the information potential, and the forces are called information forces, due to their information-theoretic origin. We create directed trees by selecting the predecessor of a node (pattern) according to the direction of the information force acting on the pattern. Each directed tree correspond to a cluster, hence enabling us to partition the data set. The clustering metric underlying our method is thus based on entropy, which is a quantity that conveys information about the shape of a probability density, and not only it's variance, as many traditional algorithms based on mere second order statistics rely on. We demonstrate the performance of our clustering technique when applied to both artificially created data and real data, and also discuss some limitations of the proposed method.

1 Introduction

In exploratory data analysis it is often desirable to partition a set of data patterns into different subsets, such that patterns within each subset are *alike* and patterns across subsets are *not alike*. This problem is known as clustering.

A wide variety of approaches to clustering have been made over the last four decades [1]. In particular, one branch of clustering techniques utilize graph theory to partition the data. Graph theoretic clustering has the advantage that parametric assumptions about data distributions do not have to be made. In

* This work was partially supported by NSF grants ECS-9900394 and EIA-0135946
** robertj@cnel.ufl.edu Phone: (+1) 352-392-2682 Fax: (+1) 352-392-0044

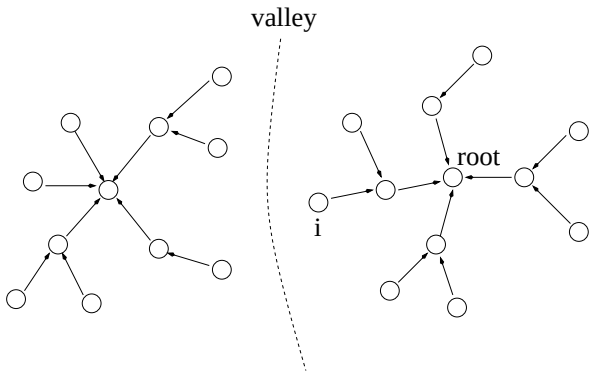


Fig. 1. Example of two directed trees, each corresponding to a cluster.

addition, normally it precludes the need to know in advance the number of clusters to be formed.

In graph theoretic clustering, usually a proximity graph [1] is constructed. In a proximity graph each node corresponds to a data pattern, which is considered a point in feature space. Between each pair of nodes an edge is formed, and the weight $d(i,j)$ on each edge is a measure of the similarity (proximity) of the nodes i and j. Clustering now becomes the problem of partitioning the proximity graph.

A common partitioning method consists of creating a hierarchy of threshold sub graphs by eliminating edges of decreasing weight in the proximity graph. Here under are included the well-known single-link [2] and complete-link [3] hierarchical clustering algorithms. Other methods form clusters by breaking inconsistent arcs in the minimum spanning tree [4] of the proximity graph, or graphs constructed from limited neighborhood sets [5]. Partitioning based on minimum cuts [6], normalized cuts [7] or variants thereof [8], have also proven efficient in clustering, especially in an image segmentation context.

A somewhat different, and less explored graph theoretic approach for detecting clusters, is based on *directed trees* [9, 10]. In a directed tree each node i initiates a branch pointing another node j, which is called the predecessor of i. Only one node does not have a predecessor, and this node is called the root. Starting from any node, the branches can be followed to the root. Note that each node except the root has one and only one predecessor, but each could be the predecessor of a number of nodes (its "children"), including zero [10].

Figure 1 shows an example of two directed trees, each corresponding to a cluster. The two clusters are separated by a valley, where the density of data points is low. Nodes near the valley, like e.g. node i, must point away from the valley in order for the clusters to be formed.

In [10] the predecessor j of node i is searched for along the steepest ascent of the probability density function, which is estimated based on points within a

local region centered around i. Node j is found within the local region, as the node closest to the steepest ascent line from i. This method is sensitive to the size of the local region, especially in the important areas near the valley.

We propose a different view to this clustering problem, derived from recent developments in information theory [11]. In our approach each data point can be considered a physical particle that experiences a force acting on it. This force is called an *Information Force* (IF), because it is the derivative (with respect to the particle) of an overall "potential energy", called the *Information Potential* (IP) [11]. The IP is defined through a non-parametric estimate of Renyi's entropy, using Parzen kernel density estimation.

For a well chosen (Gaussian) kernel size, σ, in the Parzen density estimate, the information force acting on a particle points toward the cluster the particle belongs to. This happens irrespective of the shape of the probability density describing the data set. This property can be attributed to the underlying entropy metric. The issue now is to utilize the information forces to cluster the data. It should be noted that our clustering method has resemblance to other kernel based methods, such as e.g. spectral clustering [12] and Mercer kernel based clustering [13].

Our approach is to create directed trees, each corresponding to a cluster, according to the direction of the information forces. We show that σ can also be used when selecting the predecessor node j. We search for j in a neighborhood of i, where the size of the neighborhood is specified by σ, such that j is closest to the direction of the information force acting on i.

Obviously, the parameter σ is very important. It should be determined automatically from the data at hand, and preferably each data pattern should be associated with a unique kernel, adapted based on it's neighboring data. However, at this point in time, we use only one single kernel in the Parzen estimate. The kernel size is determined manually, such that the Parzen density estimate is relatively accurate.

In the next section we explain the information-theory enabling us to define the information forces, following the outline given in [11]. In section 3 we discuss how the directed trees, which correspond to clusters, are created. We present some clustering experiments in section 4, both on artificially created data and real data. Finally, in section 5 we make our concluding remarks.

2 Information Forces

In a 1957 classic paper Jaynes [14] re interpreted statistical mechanics, providing a new viewpoint from which thermodynamic entropy and information-theory entropy [15] appear as the same concept. This advance, however, is predicated on the specifications of the data distributions.

Avoiding unrealistic parametric assumptions about data distributions, recently Principe et al. [11] combined a non-parametric density estimator with an easily computable information-theoretic definition of entropy, resulting in an entropy estimator with a very interesting physical interpretation as a potential

energy field. The entropy definition used in [11] was proposed by Renyi [16], hence called Renyi's entropy.

Renyi's entropy for a stochastic variable $\mathbf{x}$ with probability density function (pdf) $f(\mathbf{x})$ is given by [16]

$$H_R(\mathbf{x}) = \frac{1}{1-\alpha} \log \int f^{\alpha}(\mathbf{x}) d\mathbf{x}, \ \alpha > 0, \ \alpha \neq 1. \tag{1}$$

Specifically, for $\alpha = 2$ we obtain [11]

$$H_R(\mathbf{x}) = -\log \int f^2(\mathbf{x}) d\mathbf{x}, \tag{2}$$

which is called Renyi's quadratic entropy [11].

This expression can easily be estimated directly from data by the use of Parzen window density estimation, with a multidimensional Gaussian window function. We have available the set of discrete data points $\mathbf{x}_i, \ i = 1, \ldots, N$. Now, the pdf estimate based on these data points is given by [17]

$$\hat{f}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} G(\mathbf{x} - \mathbf{x}_i, \sigma^2 \mathbf{I}), \tag{3}$$

where we have used a symmetric Gaussian kernel, $G(\mathbf{x}, \mathbf{\Sigma})$, where the covariance matrix, $\mathbf{\Sigma}$, is given by $\mathbf{\Sigma} = \sigma^2 \mathbf{I}$.

By substituting (3) into (2), and utilizing the properties of the Gaussian kernel, we obtain an estimate of the entropy given by;

$$H_R(\mathbf{x}) = -\log V_R(\mathbf{x}), \tag{4}$$

where

$$V_R(\mathbf{x}) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G(\mathbf{x}_i - \mathbf{x}_j, 2\sigma^2 \mathbf{I}). \tag{5}$$

Regarding the data points $\mathbf{x}_i$ and $\mathbf{x}_j$ as physical particles, we can regard $V_{ij} = G(\mathbf{x}_i - \mathbf{x}_j, 2\sigma^2 \mathbf{I})$ as an interaction law between particles, imposed by the Gaussian kernel. This interaction law is always positive and is inversely proportional to the distance between particles.

The sum of interactions on the i'th particle is $V_i = \sum_j V_{ij}$. The sum of all pairs of interactions, given by (5), can now be regarded as an overall potential energy of the data set, where the local field strength between pairs of particles is governed by the width of the Gaussian kernel [11]. This potential energy is called the information potential.

Just as in mechanics, the force acting on particle $\mathbf{x}_i$ is given by the derivative of the potential field with respect to the particle;

$$\begin{aligned}\mathbf{F}_i &= \frac{\partial}{\partial \mathbf{x}_i} V_R(\mathbf{x}) \\ &= -\frac{1}{N^2\sigma^2} \sum_{j=1}^{N} G(\mathbf{x}_i - \mathbf{x}_j, 2\sigma^2\mathbf{I})(\mathbf{x}_i - \mathbf{x}_j) \\ &= -\frac{1}{N^2\sigma^2} \sum_{j=1}^{N} V_{ij}\mathbf{d}_{ij}, \end{aligned} \tag{6}$$

where $\mathbf{d}_{ij} = \mathbf{x}_i - \mathbf{x}_j$. This is the net effect of the IP on particle $\mathbf{x}_i$, and will be called an information force.

Naturally, the behavior of the information forces and the quality of the pdf estimate inherent in the entropy estimator, are closely related. However, our concern at this stage is mainly whether a particle $\mathbf{x}_i$ experiences a force pushing it toward a cluster or not. Whether the pdf estimate is the most accurate possible, is of lesser concern.

In Fig. 2 (a) a data set consisting of three elongated clusters and one spherical cluster is shown, and the IF acting on each data point is indicated by an arrow. The arrows only convey information about the directions of the forces, not the magnitude. Before calculating the IFs the data set was normalized feature-by-feature to lie in a range $[-1, 1]$. A kernel size of $\sigma = 0.03$ was used in the pdf estimation. It can be seen that nearly all the IFs point inward to one of the clusters. A few outliers mostly interact with each other, because the kernel size is small. The corresponding pdf estimate is shown in Fig. 2 (c). This is a rather crude and noisy estimate, indicating that if our concern is solely density estimation, σ is probably too low.

Figure 2 (b) and (d) show the IFs and the corresponding pdf estimate for $\sigma = 0.09$. The IFs point inward to one of the clusters also in this case. The pdf estimate clearly shows the structure of the data, but the increasingly dominant smoothing effect resulting from a large kernel size is evident.

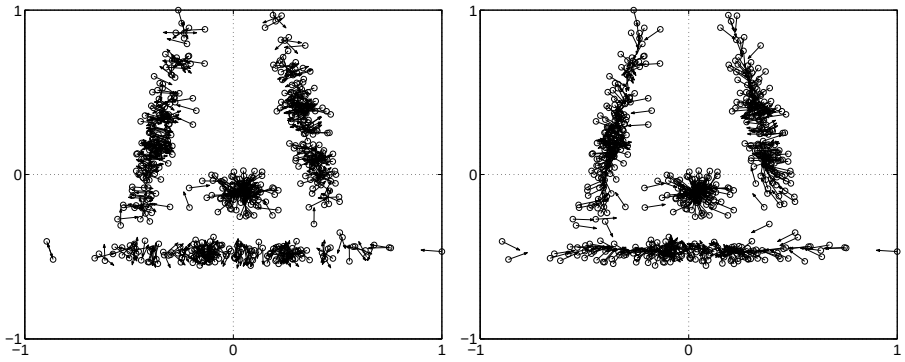
A close look at Fig. 2 (a) and (b) also shows that some of the IFs points to different clusters for the two different kernel sizes.

3 Creating Directed Trees

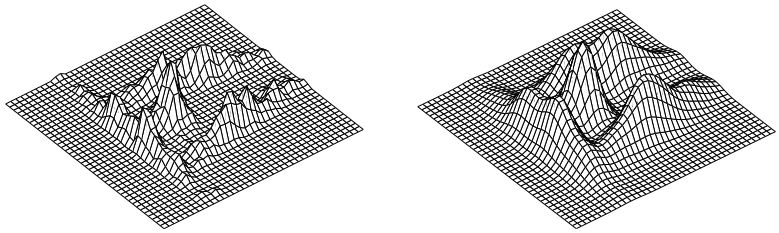
Our procedure for creating directed trees is very simple, once the IFs have been calculated. We examine every data point $\mathbf{x}_i$, $i = 1, \ldots, N$, one at a time, where $\mathbf{x}_i$ corresponds to node i in the final tree. For node i we determine whether it has a predecessor j, or whether it is a root, based on the following: Node j is the predecessor of node i if it satisfies

- Node j lies *closest to the direction of the force* $\mathbf{F}_i$ *acting on* i,

under the following constraints;



(a) IFs, $\sigma = 0.03$ (b) IFs, $\sigma = 0.09$



(c) pdf, $\sigma = 0.03$ (d) pdf, $\sigma = 0.09$

Fig. 2. (a) and (b): Example of a data set and the IFs acting on each particle for two different values of σ. (c) and (d): The corresponding Parzen pdf estimates.

1. The distance $\mathbf{x}_i - \mathbf{x}_j \leq 3\sigma$.
2. $\mathbf{F}_i \cdot (\mathbf{x}_i - \mathbf{x}_j) \geq 0$.
3. Node j can not be one of i's children.

If there exists no node j satisfying the above constraints, then node i is defined to be a root, not pointing to another node.

The only free parameter, σ, is the same as the one already used when determining the IFs. The end result of this procedure is a set of directed trees, each corresponding to a cluster.

Constraint 1 is necessary in order to avoid linking together trees that are in fact part of different clusters. Consider Fig. 3. The tiny arrows show how nodes have been connected up to a point in time. The nodes with big arrows have not yet been examined, and the arrows show the direction of the IF acting on each

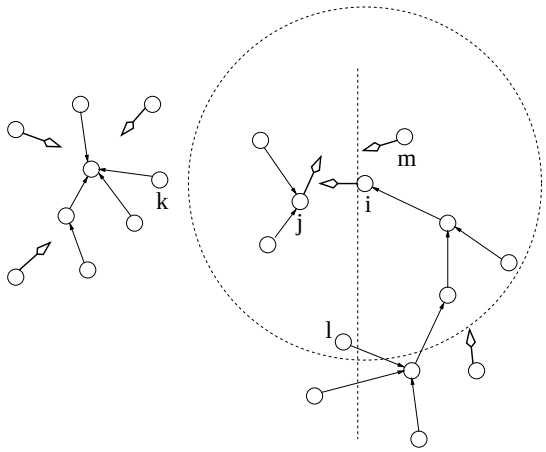


Fig. 3. Creating directed trees.

one of them. Let us examine node i. Of all the nodes pointing inward to the cluster i belongs to, node j is closest to the direction of $\mathbf{F}_i$. However, node k, actually belonging to a different cluster, is even closer to the direction of $\mathbf{F}_i$. To avoid k being selected as the predecessor of i, we must restrict our search to a neighborhood of node i. We find that simply defining the neighborhood of i to be given by a hyper-sphere of radius 3σ centered at i, is reasonable. In Fig. 3 the neighborhood of node i is indicated by the dashed circle.

Our choice of neighborhood is reasonable based on the inherent properties of Parzen pdf estimation. In order for the Parzen pdf estimate to be relatively accurate, the Gaussian kernels must be chosen such that the effective width of the kernels is sufficiently large, but not too large. If it is sufficiently large, a kernel centered at a data point also covers several other of its neighboring data points. But it doesn't cover distant data points, because that would result in to smooth an estimate. The effective width of a Gaussian kernel is given by 3σ, since 98% of its power is concentrated within 3σ of its center.

Constraint 2 is crucial, since it ensures that we really use the information provided by the direction of the IFs. Figure 3 illustrates this point. Node m, or any of the other nodes to the right of the dashed line, is not allowed to be selected as predecessor of node i. The reason for this is obvious, since the whole idea of our clustering technique is to use the IFs to create directed trees since the IFs point toward clusters, and not away from them.

Constraint 3 ensures that a node do not become one of its own children, contradicting the idea of a directed tree. For instance, in Fig. 3, node l can not be a predecessor of node i, even though it is located within the neighborhood of i.

4 Clustering Experiments

We present some clustering experiments, both on an artificially created data set, and two real data sets. In all experiments the data have been normalized feature-by-feature to have a range $[-1, 1]$.

We create the directed trees, and for each tree assign the same label to its members. Outliers in the data set will tend to create trees with only one or a few members. Clusters with 5 members or less are kept in an outlier set, and are not assigned a label. Labeling these points can be done in a post-clustering operation, for example by simple nearest-neighbor classification.

4.1 Artificial Data Sets

First, we re-visit the data set considered in Fig. 2. Figure 4 (a) shows the same data set, where the data points belonging to the same cluster have been marked by the same symbol. It can be seen that three of the clusters are elongated, with a shape making this data set very difficult for variance based clustering methods.

Figure 4 (b) shows the clustering result our IF directed tree method produces when applied to this data set, for a kernel size $\sigma = 0.07$. The result is satisfying. Only three patterns have been assigned to the wrong cluster, and there is one outlier. Furthermore, the outlier would be assigned to the correct cluster after a nearest-neighbor classification.

Figure 5 shows the clustering results for a range of σ's. For $\sigma = 0.06$ there are three errors, and three outliers. When the kernel size increases the number of outliers decreases, while the number of errors increases as σ increases.

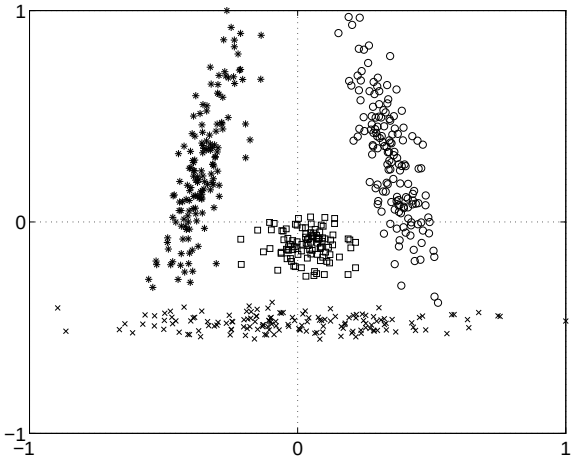
For $\sigma > 0.085$ our experiments show that clusters tend to be merged together across cluster boundaries. For $\sigma < 0.06$ the clusters tend to be split. E.g. for $\sigma = 0.055$ one of the clusters is split into two clusters. Even though clusters are split, the method can still be useful if a merging procedure is implemented.

However, we see that even though we previously have shown that the IFs point inward to clusters for at least $0.03 \leq \sigma \leq 0.09$, the overall clustering procedure is somewhat more sensitive to the size of σ.

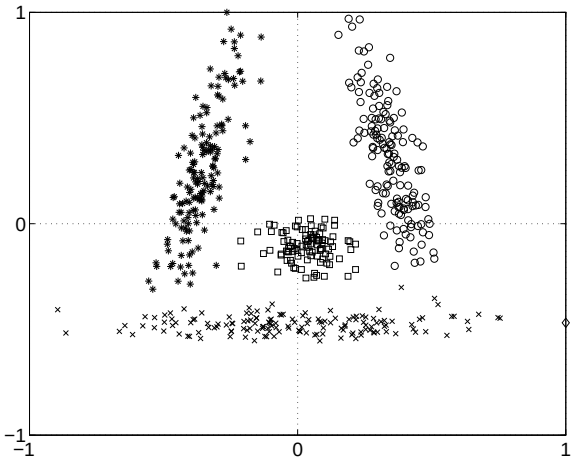
For comparison we show in Fig. 6 the clustering result the K-means [18] algorithm produces on the same data set. Since the K-means algorithm has a tendency to be trapped in local minima when minimizing the K-means cost, we have shown the best result out of 10 runs. Since K-means is based on a minimum variance criterion, it only works well for hyper-spherical, or at best hyper-elliptical data. This can be clearly observed, as K-means fails on our data set consisting of several elongated clusters.

The second artificially created data set consists of two highly irregular clusters. On this data set we are able to produce a perfect clustering for a kernel size in the range $0.1 \leq \sigma \leq 0.12$. Figure 7 (a) shows the result of our method for $\sigma = 0.1$.

In Fig. 7 (b) the result for K-means is shown. Again, K-means fails, since it can not handle clusters having a non-linear boundary between them.



(a) True labels.



(b) Result of IF directed tree labeling. $\sigma = 0.07$.

Fig. 4. Two-dimensional artificially created data set used in clustering experiment.

4.2 Real Data Sets

Next, we test our method on the WINE data set, extracted from the UCI repository database [19]. This data set consists of 178 instances in a 13-dimensional feature space, where the features are found by chemical analysis of three different types of wines. We include this data set in our analysis, because it shows

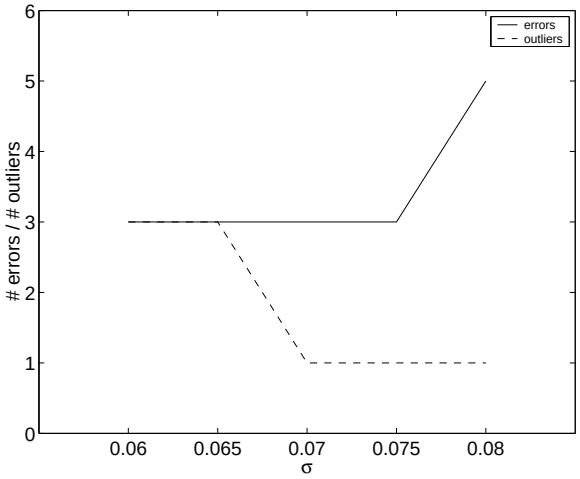


Fig. 5. # of errors and # of outliers for two-dimensional data set plotted as a function of σ.

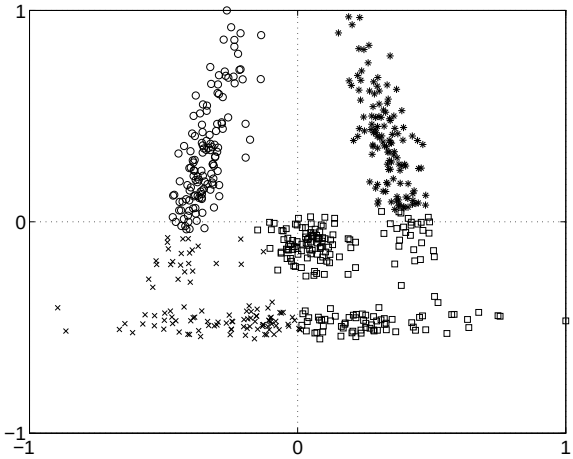
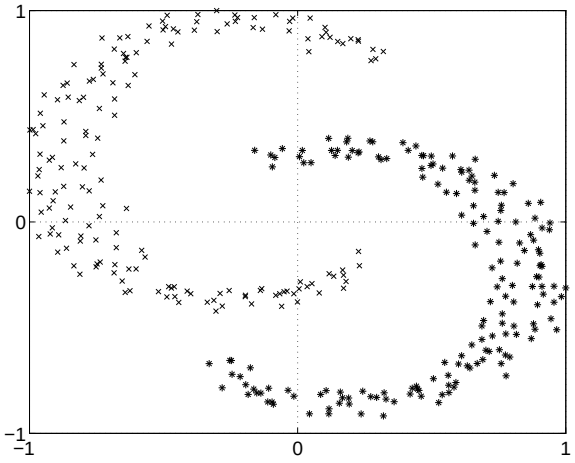


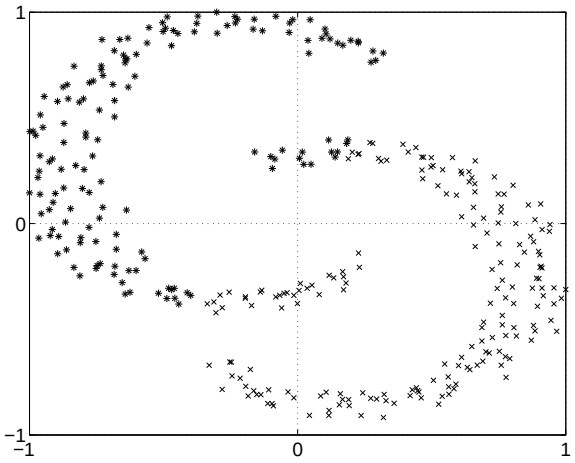
Fig. 6. Result of K-means clustering.

that our clustering method is capable of performing well in a high dimensional feature space.

For a range $0.29 \leq \sigma \leq 0.32$ we obtain satisfactory clustering results. The confusion matrix using $\sigma = 0.32$ is shown in Table 1. We denote each class by $C_i,\ i = 1, 2, 3$. The numbers in parenthesis indicate the number of instances actually belonging to each class. Table 1 shows that there are 10 patterns as-



(a) Result of IF directed tree labeling for highly irregular clusters. $\sigma = 0.1$.



(b) Result of K-means clustering.

Fig. 7. Two-dimensional artificially created data set used in clustering experiment.

signed to the wrong class. The 10 patterns in fact belong to C_2, but some are assigned to C_1, and some to C_3. There are a total of 13 outliers. Again C_2 is the troublesome class, since 11 of the outliers belong to this class. We have not made any attempt to classify the outliers by other means, so we do not know whether

Table 1. Confusion matrix for wine data; $\sigma = 0.32$.

			Result		
			C_1	C_2	C_3
True	C_1	(59)	59	0	0
	C_2	(71)	4	50	6
	C_3	(48)	0	0	46

they would be assigned to class C_2 or not in a post-clustering operation. The last two outliers belong to C_3.

Last, we have included an experiment clustering the well-known IRIS data set, also extracted from the UCI repository database. This data set contains three classes of 50 patterns each, where each class refers to a type of iris plant. It is characterized by four numeric attributes. The IRIS data set is known to be difficult to cluster, since two of the classes overlap to some degree, and the boundaries between classes are non-linear. In this case, our method is expected to have difficulty in producing a satisfying clustering result. When there is no clear boundary between clusters, the particles of the overlapped clusters will interact more with each other, and the information forces will seize to point toward the distinct clusters.

Table 2. Confusion matrix for iris data; $\sigma = 0.095$.

			Result		
			C_1	C_2	C_3
True	C_1	(50)	49	0	0
	C_2	(50)	0	42	3
	C_3	(50)	0	5	44

Consequently, in this case our clustering method is more sensitive to the kernel size than in the other experiments. The range of σ's for which we obtain reasonable results is narrow. However, choosing e.g. $\sigma = 0.095$, we obtain the confusion matrix shown in Table 2. In this case eight patterns are assigned to the wrong class. The incorrect labeled patterns clearly belong to the two clusters that overlap somewhat. In addition there are seven outliers.

5 Conclusion

We have presented a new graph-theoretic clustering method. The core idea is to utilize the information forces to create directed trees, each corresponding to a cluster.

The directed trees are created by searching for the predecessor of node i, in the direction of the information force acting on i. The information force can be regarded as a force acting on a physical particle, imposed by an overall potential energy. The potential energy is called the information potential, because of its information-theoretic origin.

The information forces contain global information about all data points. Hence, we avoid having to rely on local estimates of the gradient of the pdf, which is known to be sensitive to the size of the local region, especially in the important areas near the valleys of the data distribution.

A non-parametric estimator of Renyi's entropy is obtained via Parzen (Gaussian) kernel density estimation. The information potential is based on this estimator, and hence the information forces too. The main advantage of our approach is that the underlying clustering metric is based on entropy, which is a quantity that conveys information about the shape of a probability density, and not only it's variance, which many traditional clustering algorithms, e.g. K-means, rely on.

To determine the information forces a single $O(N^2)$ operation is required. Determining the neighbors of each pattern requires a further $O(N)$ search procedure in each case. The $O(N^2)$ operation is computationally demanding for large data sets. Still, our clustering method is less computationally expensive than a previous iterative clustering algorithm proposed by Gokcay and Principe [20] based on the same theory underlying the Renyi entropy estimator.

At present, the major problem with our clustering algorithm is how to choose the kernel size σ. This is a problem encountered in all kernel based methods, both unsupervised and supervised. For our current clustering method to be useful in practice, an automatic procedure must be implemented, determining σ such that the Parzen pdf estimate is relatively accurate. Preferably, each data pattern $\mathbf{x}_i$ should be associated with a unique kernel, σ_i, adapted based on it's neighboring data. In regions of high density of data patterns the kernels should be relatively narrow, and in regions of low density the kernels should be relatively wide. The use of non-symmetric kernels, may also potentially help. The σ_i's could in this case e.g. be determined based on the covariance matrix estimated in a neighborhood of $\mathbf{x}_i$.

In our current method, it is also possible to avoid having to define the neighborhood of a pattern directly in terms of σ when creating the directed trees. Instead, the neighborhood of $\mathbf{x}_i$ could for example be defined in terms of the inverse of the magnitude of the IF acting on it. The magnitude of the IFs will be relatively large in areas of high density of data patterns, and relatively small in areas of low density. Decoupling the creation of the directed trees from σ could lead to improvements, since we have found that our algorithm may be more dependent on the value of σ when creating the directed trees, compared to it's effect on the actual information forces.

For clusters having a large degree of overlap, our current method will encounter increasing difficulty, because the information forces no longer points to

distinct cluster centers. This was observed when trying to cluster the IRIS data set. The range of σ's giving satisfying results was narrow.

We have performed several clustering experiments both on an artificially created data set and two real data sets. We have shown that our method has the ability to discover clusters of irregular shape, also in a high dimensional feature space.

Finally, to put the clustering results of our graph theoretic information force clustering algorithm in some perspective, we compare the result we obtained on the most difficult data set for our algorithm to handle, namely the IRIS data set, with the results achieved by some other recent clustering methods on the same data set. The results are not straightforward to compare, because the data are often pre-processed in different ways. As an example Eltoft and deFigueiredo [21] reported on the average six errors (or outliers), where clustering is based on the first and second principal components. Gokcay and Principe [20] obtained 14 errors when normalizing the data the same way we did. Ben-Hur et al. [22] report four errors when the clustering is based on the first three principal components, and 14 when the fourth principal component is added to the feature vectors. The information-theoretic approach of Tishby and Slonim [23] leads to five mis classifications and the SPC algorithm of Blatt et al. [24], when applied to the original data space, has 15 mis classifications. Horn and Gottlieb [25] report five errors when the data is normalized in a certain manner. Without this normalization they obtained 15 errors. The best result to our knowledge is the kernel-based method of Girolami [13], obtaining only three partition errors.

References

1. A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
2. P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy*, Freeman, London, 1973.
3. B. King, "Step-wise clustering procedures," *J. Am. Stat. Assoc.*, pp. 86–101, 1967.
4. C. T. Zahn, "Graph Theoretic Methods for Detecting and Describing Gestalt Clusters," *IEEE Trans. Comput.*, vol. 20, pp. 68–86, 1971.
5. R. Urquart, "Graph Theoretical Clustering based on Limited Neighborhood Sets," *Pattern Recognition*, vol. 15, pp. 173–187, 1982.
6. 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*, vol. 15, no. 11, pp. 1101–1113, 1993.
7. 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.
8. C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A Min-max Cut Algorithm for Graph Partitioning and Data Clustering," in *IEEE Int. Conf. on Data Mining*, 2001, pp. 107–114.
9. W. L. G. Koontz, P. M. Narendra, and K. Fukunaga, "A graph-theoretic approach to nonparametric cluster analysis," *IEEE Transactions on Computers*, vol. 25, pp. 936–944, 1975.

10. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, New York, 1990.
11. J. Principe, D. Xu, and J. Fisher, *Unsupervised Adaptive Filtering*, vol. 1, chapter 7 "Information Theoretic Learning", John Wiley & Sons, 2000.
12. A. Y. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," in *Advances in Neural Information Processing Systems*, 2002, number 14, pp. 849–856.
13. M. Girolami, "Mercer Kernel-Based Clustering in Feature Space," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 780–784, 2002.
14. E. T. Jaynes, "Information Theory and Statistical Mechanics," *The Physical Review*, vol. 106, no. 4, pp. 620–630, 1957.
15. C. E. Shannon, "A Mathematical Theory of Communication," *Bell Sys. Tech. J.*, vol. 27, pp. 379–423, 623–653, 1948.
16. A. Renyi, "On Measures of Entropy and Information," in *Fourth Berkeley Symposium on Mathematical Statistics and Probability*, 1960, pp. 547–561.
17. E. Parzen, "On the Estimation of a probability density function and the mode," *Ann. Math. Stat.*, vol. 32, pp. 1065–1076, 1962.
18. J. McQueen, "Some methods for classification and analysis of multivariate observations," in *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.
19. R. Murphy and D. Ada, "UCI Repository of Machine Learning databases," Tech. Rep., Dept. Comput. Sci. Univ. California, Irvine, 1994.
20. E. Gokcay and J. Principe, "Information Theoretic Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 158–170, 2002.
21. T. Eltoft and R. J. P. deFigueiredo, "A New Neural Network for Cluster-Detection-and-Labeling," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 1021–1035, 1998.
22. A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support Vector Clustering," *Journal of Machine Learning Research*, vol. 2, pp. 125–137, 2001.
23. N. Tishby and N. Slonim, "Data Clustering by Markovian Relaxation and the Information Bottleneck Method," in *Advances in Neural Information Processing Systems*, Denver, USA, 2000, vol. 13, pp. 640–646.
24. M. Blatt, S. Wiseman, and E. Domany, "Data Clustering using a Model Granular Magnet," *Neural Computation*, vol. 9, no. 8, pp. 1805–1842, 1997.
25. D. Horn and A. Gottlieb, "The Method of Quantum Clustering," in *Advances in Neural Information Processing Systems*, Vancouver, Canada, 2001, vol. 14, pp. 769–776.