

Article

Information Theoretic Hierarchical Clustering

Mehdi Aghagolzadeh ^{1,2}, Hamid Soltanian-Zadeh ^{1,3,4,*} and Babak Nadjar Araabi ^{1,4}

¹ Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, PO Box 1439957131, Tehran 14395-515, Iran

² Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA; E-Mail: aghagolz@msu.edu

³ Radiology Image Analysis Laboratory, Henry Ford Health System, Detroit, MI 48202, USA

⁴ School of Cognitive Sciences, Institute for Research in Fundamental Sciences (IPM), PO Box 1954856316, Tehran, Iran; E-Mail: araabi@ut.ac.ir

* Author to whom correspondence should be addressed; E-Mail: hamids@rad.hfh.edu.

Received: 8 December 2010; in revised form: 31 December 2010 / Accepted: 27 January 2011 / Published: 10 February 2011

Abstract: Hierarchical clustering has been extensively used in practice, where clusters can be assigned and analyzed simultaneously, especially when estimating the number of clusters is challenging. However, due to the conventional proximity measures recruited in these algorithms, they are only capable of detecting mass-shape clusters and encounter problems in identifying complex data structures. Here, we introduce two bottom-up hierarchical approaches that exploit an information theoretic proximity measure to explore the nonlinear boundaries between clusters and extract data structures further than the second order statistics. Experimental results on both artificial and real datasets demonstrate the superiority of the proposed algorithm compared to conventional and information theoretic clustering algorithms reported in the literature, especially in detecting the true number of clusters.

Keywords: information theory; Rényi's entropy; quadratic mutual information; hierarchical clustering; proximity measure

1. Introduction

Clustering is an unsupervised approach for segregating data into its natural groups, such that the samples in each group have the highest similarity with each other and the highest dissimilarity with samples of the other groups. Clustering is in general exploited when labeling data by a human operator is expensive and subject to error, and has many applications in data mining, image segmentation, remote sensing, and compression, to name but a few.

One of the main characteristics of any clustering algorithm is its definition of proximity measure. Various clustering algorithms have different notions of proximity. For instance, measures such as the Euclidean distance or the within cluster variance, also referred to as the Mahalanobis distance [1], can explore up to the second order statistics of the data. Determining an appropriate proximity measure for clustering is a challenging task that directly depends on the structure of the data. With an ill-defined proximity measure, even compatible clustering algorithms fail in accurately identifying the data structures. Since conventional clustering algorithms exploit within cluster variance measures, they are only capable of identifying spherical mass-shaped clusters, while complicated shapes are disregarded. Algorithms such as k -means [2], fuzzy c -means [3], divisive and agglomerative hierarchical clustering [4] fall into this category. On the other hand, clustering based on artificial neural networks [5] and support vector machines [6] can identify clusters with various shapes, however, are computationally expensive and require perfect tunings.

Information theoretic measures have been proposed as proximity measures that can extract data structures further than the second order statistics [7,8]. However, practical difficulties in estimating the distribution of data have significantly reduced the applicability of such proximity measures in clustering, especially when no prior information about the data structures is given. The distribution can be estimated by either parametric models, such as a mixture of Gaussian functions [9], or by non-parametric models, such as the Parzen window estimator [10]. Regardless of the model used for estimating the distribution, the performance of any information theoretic clustering totally depends on how well the estimator predicts this distribution, its computational cost, and its ability in updating the distribution as the clustering proceeds.

Information theoretic clustering algorithms have tackled the challenges of estimating the distribution from different prospects. Mutual information, defined using either the Shannon's or Kolmogorov's interpretation of information, has been used for combining clusters in an agglomerative hierarchical clustering, in which the distribution is approximated using the k -nearest neighbor estimator [11]. Using the grid and count method for estimating the distribution, the statistical correlation among clusters was minimized for clustering gene-wide expression data [12]. The k -nearest neighbor estimator is sub-optimal in the sense that it requires re-estimating the distribution after the clusters are updated. Although the grid and count method benefits from updating the distribution of combined clusters from the existing ones, but this algorithm produces biased estimations for small-sized high-dimensional data.

In this paper, the distribution is estimated using a Parzen window estimator with Gaussian kernels centered on each sample and with a constant covariance. Although this distribution seems superficial and computationally expensive, but exploiting the Rényi's entropy estimator [13] in a quadratic form as the proximity measure, the mutual information can be estimated from pairwise distances, also

referred to as the quadratic mutual information [14]. This proximity measure has been used in an iterative clustering to optimize the clustering evaluation function that will find the nonlinear boundaries between clusters [15]. It also has been used in learning the discriminant transform from the mutual information estimated between the cluster labels and the transformed features for classification [16]. Here, we use a similar technique in finding the association between the data samples and cluster labels using the quadratic mutual information. We will show how increasing the quadratic mutual information assigns appropriate clusters to the data.

First, by introducing the clustering as a distortion-rate problem, we will show how optimizing the distortion-rate function provides us with the best clustering result. We propose a hierarchical approach for this optimization. Unlike partitioning approaches such as k -means clustering that primarily require setting the number of clusters, the hierarchical approach gives us the additional ability of detecting the number of clusters in data, especially when no prior information is available. Starting from an initial set of clusters generated by a simple clustering algorithm, in each hierarchy, a cluster is eliminated and merged with the remaining clusters, until one cluster remains. Eventually, based on the variations in the mutual information, the true number of clusters is determined.

We propose two algorithms for the hierarchical optimization, the agglomerative and the split-and-merge clustering. In the former, at any hierarchy, the two clusters that maximize the mutual information are combined into one cluster. In the latter one, a cluster that has the worst effect on the mutual information is singled out for elimination. This cluster is split and its samples are allocated to the remaining clusters. Both these methods maximize the mutual information and have advantages compared to one another. In the following section, we will first demonstrate how optimizing the distortion-rate function provides us with the best clustering and then show how the mutual information can be approximated by the quadratic mutual information. The two hierarchical approaches are also demonstrated in this section. In Section 3, we will demonstrate the performance of the proposed hierarchical clustering on both artificial and real datasets, and will compare them with clustering algorithms reported in the literature.

2. Theory

2.1. Distortion-Rate Theory

Clustering can be viewed as projecting a large number of discrete samples from the input space, into a finite set of discrete symbols in the clustered space, where each symbol resembles a cluster. Thus, clustering is a many-to-one mapping from the input space, X , to the clustered space, \hat{X} , and can be fully characterized by the conditional probability distribution, $p(\hat{x}|x)$. Using this mapping, the distribution of the clustered space is estimated as:

$$p(\hat{x}) = \sum_{x \in X} p(x)p(\hat{x}|x) \quad (1)$$

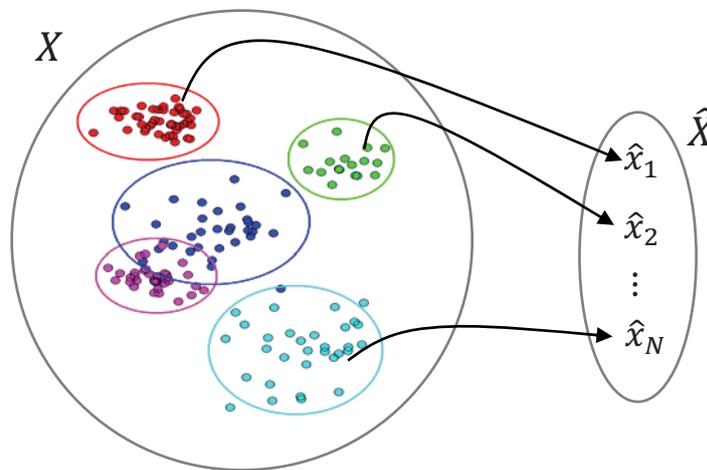
where $p(x)$ is the distribution of the input space. Figure 1 demonstrates a many-to-one mapping, where each symbol, $\hat{x}_i \in \hat{X}$, for $i = 1, 2, \dots, N$, represents a cluster of samples from the input space, and N is the number of clusters.

Although clusters have different number of samples, but the average number of samples in each cluster is $2^{H(\hat{X}|X)}$, where $H(\hat{X}|X)$ is the conditional entropy of the clustered space given the input space and is estimated as:

$$H(\hat{X}|X) = \sum_{x \in X} \sum_{\hat{x} \in \hat{X}} p(x)p(\hat{x}|x) \log p(\hat{x}|x) \tag{2}$$

The number of clusters is $2^{H(\hat{X})}$, where $H(\hat{X})$ is the entropy of the clustered space. Note that $H(\hat{X})$ is upper bounded by $\log N$ and is equal to the upper bound only when all clusters have an equal number of samples.

Figure 1. Demonstration of a many-to-one mapping from the input space, including semi-infinite number of discrete samples, to a finite number of symbols, N , in the clustered space.



To obtain a lossless many-to-one mapping, the immediate goal is to preserve the information in X in the projected space, \hat{X} . The loss of information due to mapping is measured by the conditional entropy, $H(X|\hat{X}) = H(X) - I(X; \hat{X})$, where $H(X)$ is the amount of information in X . The mutual information between the input and clustered space, $I(X; \hat{X})$, is estimated as:

$$I(X; \hat{X}) = \sum_{x \in X} \sum_{\hat{x} \in \hat{X}} p(x, \hat{x}) \log \frac{p(x, \hat{x})}{p(x)p(\hat{x})} = \sum_{x \in X} \sum_{\hat{x} \in \hat{X}} p(x)p(\hat{x}|x) \log \frac{p(\hat{x}|x)}{\sum_{x \in X} p(x)p(\hat{x}|x)} \tag{3}$$

Notice how the mutual information is estimated based on only the input distribution, $p(x)$, and mapping distribution, $p(\hat{x}|x)$. Mutual information gives us the rate by which the clustered space represents the input space. For a lossless mapping, $H(X|\hat{X}) = 0$ or $I(X; \hat{X}) = H(X)$, which in turn means that all the information in the input space is sent to the clustered space. While a higher rate for the clustered space generates less information loss, reducing this rate increases the information loss, therefore introducing a tradeoff between the rate and the information loss. In clustering, the goal is to introduce a lossy many-to-one mapping that reduces the rate by representing the semi-infinite input space with a finite number of clusters, thus introducing information loss such that $I(X; \hat{X}) \leq H(X)$.

The immediate goal in clustering is to introduce clusters with the highest similarity or lowest distortion among its samples. Distortion is the expected value of the distance between the input and clustered spaces, $d(x, \hat{x})$, defined based on the joint distribution, $p(x, \hat{x})$, as:

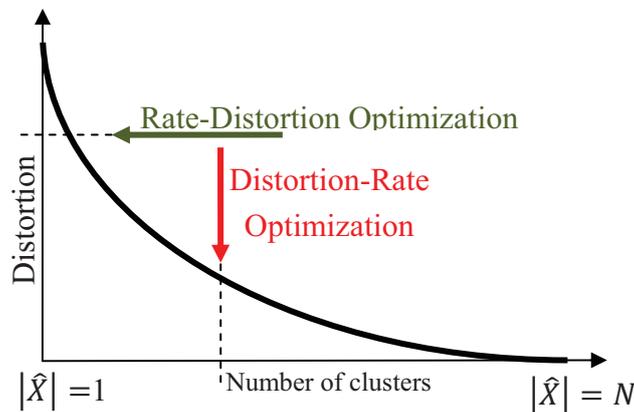
$$E_{p(x,\hat{x})}[d(x,\hat{x})] = \sum_{x \in X} \sum_{\hat{x} \in \hat{X}} p(x,\hat{x})d(x,\hat{x}) \tag{4}$$

Different proximity measures can be defined as distortion, for instance, for the Euclidean distance, \hat{x} are the center of clusters and $d(x,\hat{x}) = \sum_{x \in X} (x - \hat{x})^2$ is the cluster variance and $p(x,\hat{x}) = \frac{1}{Z} I(x \in \hat{x})$ is a uniform distribution, where $Z = \sum_{\hat{x} \in \hat{X}} p(x,\hat{x})$ is the normalizing term and $I(\cdot)$ is the indicator function. The tradeoff between the preserved amount of information and the expected distortion is characterized by the Shannon-Kolmogorov rate-distortion function, where the goal is to achieve the minimum rate for a given distortion, illustrated by the horizontal arrow in Figure 2. The rate-distortion optimization has been extensively used for quantization, where the goal is to achieve the minimum rate for a desired distortion [17]. Unlike quantization, the goal in clustering is to minimize the distortion for a preferred number of clusters, N , thus, the distortion-rate function is optimized instead:

$$D(R) = \min_{I(X;\hat{X}) \leq R} E[d(x,\hat{x})] \tag{5}$$

In Figure 2, the vertical arrow demonstrates the distortion-rate optimization that achieves the lowest distortion for a desired rate. Note that the number of clusters, N , places an upper bound on the rate, determined by the mutual information. Assuming that decreasing distortion monotonically increases the mutual information, clustering can be interpreted as maximizing the mutual information for a fixed number of clusters, $\hat{X} = \max_{|\hat{X}|=N} I(X;\hat{X})$, where $|\hat{X}|$ is the number of clusters.

Figure 2. Demonstration of the rate-distortion and distortion-rate optimizations by the horizontal and vertical arrows, respectively.



2.2. Quadratic Mutual Information

The Shannon’s mutual information estimates the distance between the joint distribution, $p_{XY}(x,y)$, and the product of the marginal distributions, $p_X(x)p_Y(y)$, [18], as:

$$I(X;Y) = \int \int p_{XY}(x,y) \log \frac{p_{XY}(x,y)}{p_X(x)p_Y(y)} dx dy \tag{6}$$

The mutual information has also been referred to as the Kullback-Leibler divergence between $p_{XY}(x,y)$ and $p_X(x)p_Y(y)$ [19]. Due to the challenges in estimating the Shannon’s mutual information,

the Euclidean distance between $p_{XY}(x, y)$ and $p_X(x)p_Y(y)$ can be used instead as an approximation for mutual information, $\tilde{I}(X; Y)$, also referred to as the quadratic divergence between distributions [20]:

$$\begin{aligned} \tilde{I}(X; Y) &= \int \int_{X \ Y} (p_{XY}(x, y) - p_X(x)p_Y(y))^2 dx dy \\ &= \int \int_{X \ Y} p_{XY}^2(x, y) dx dy - 2 \int \int_{X \ Y} p_{XY}(x, y)p_X(x)p_Y(y) dx dy + \int p_X^2(x) dx \int p_Y^2(y) dy \end{aligned} \tag{7}$$

Considering the quadratic Rényi’s entropy estimator, $H_2(X) = -\ln \int f^2(x) dx$, this entropy also includes the quadratic form of the distribution, $f(x)$. Note that the quadratic Rényi’s entropy estimator is the second order, $\alpha = 2$, of the Rényi’s entropy estimator, $H_\alpha(X) = \frac{1}{1-\alpha} \ln \int f^\alpha(x) dx$ [13]. By disregarding the logarithm in the quadratic Rényi’s entropy, the quadratic approximation of mutual information in (7) is a valid estimator of the information content, and indeed can be used for clustering.

2.3. Parzen Window Estimator with Gaussian Kernels

The distribution of samples in cluster \mathcal{X} is approximated by the non-parametric Parzen window estimator with Gaussian kernels [21,22], in which a Gaussian function is centered on each sample as:

$$p_X(x) = \frac{1}{|\mathcal{X}|} \sum_{i=1}^{|\mathcal{X}|} \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x - x_i)^T \Sigma^{-1}(x - x_i)\right) \tag{8}$$

where T is transpose, d is the dimension of x , Σ is the covariance matrix, $x_i \in \mathcal{X}$ are the samples of cluster \mathcal{X} , and the cardinality $|\mathcal{X}|$ is the number of samples in that cluster. Assuming the variances for different dimensions are equal and independent from each other, thus, providing us with a diagonal covariance matrix with constant elements, δ_X^2 , the distribution is simplified as:

$$p_X(x) = \frac{1}{|\mathcal{X}|} \sum_{i=1}^{|\mathcal{X}|} \frac{1}{\sqrt{(2\pi\delta_X^2)^d}} \exp\left(-\frac{(x - x_i)^2}{2\delta_X^2}\right) = \frac{1}{|\mathcal{X}|} \sum_{i=1}^{|\mathcal{X}|} \mathcal{N}(x - x_i, \delta_X^2) \tag{9}$$

where $\mathcal{N}(\mu, \delta^2)$ is a Gaussian function with mean μ and variance δ^2 . Using the distribution estimator in (9), the quadratic terms in (7) can be further simplified as:

$$\begin{aligned} \int_X p_X^2(x) dx &= \frac{1}{|\mathcal{X}_k|} \frac{1}{|\mathcal{X}_l|} \sum_{i=1}^{|\mathcal{X}_k|} \sum_{j=1}^{|\mathcal{X}_l|} \int \mathcal{N}(x - x_i, \delta_X^2) \mathcal{N}(x - x_j, \delta_X^2) dx \\ &= \frac{1}{|\mathcal{X}_k|} \frac{1}{|\mathcal{X}_l|} \sum_{i=1}^{|\mathcal{X}_k|} \sum_{j=1}^{|\mathcal{X}_l|} \mathcal{N}(x_i - x_j, 2\delta_X^2) \end{aligned} \tag{10}$$

where $x_i \in \mathcal{X}_k$ and $x_j \in \mathcal{X}_l$ are the samples from clusters \mathcal{X}_k and \mathcal{X}_l , respectively, and $\{\mathcal{X}_k, \mathcal{X}_l\} \in \hat{\mathcal{X}}$ are clusters from the clustered space. Note that the convolution of two Gaussian functions is also a Gaussian function.

Back to the clustering problem, in which the input space is the individual samples and clustered space is the finite number of clusters, the quadratic mutual information in (7) is restructured in the following discrete form:

$$\begin{aligned} \tilde{I}(X; \hat{X}) &= \sum_{x \in X} \sum_{\hat{x} \in \hat{X}} (p_{X\hat{X}}(x, \hat{x}) - p_X(x)p_{\hat{X}}(\hat{x}))^2 \\ &= \sum_{x \in X} \sum_{\hat{x} \in \hat{X}} p_{X\hat{X}}^2(x, \hat{x}) - 2 \sum_{x \in X} \sum_{\hat{x} \in \hat{X}} p_{X\hat{X}}(x, \hat{x})p_X(x)p_{\hat{X}}(\hat{x}) + \sum_{x \in X} \sum_{\hat{x} \in \hat{X}} p_X^2(x)p_{\hat{X}}^2(\hat{x}) \end{aligned} \tag{11}$$

The distribution of data, $p_X(x)$, is equal to the distribution of all samples considered as one cluster, and is estimated using (9) as:

$$p_X(x) = \frac{1}{|X|} \sum_{i=1}^{|X|} \mathcal{N}(x - x_i, \delta_X^2) \tag{12}$$

where $|X|$ is the total number of samples, $|X| = \sum_{k=1}^N |\mathcal{X}_k|$, and $|\mathcal{X}_k|$ is the number of samples in the k^{th} cluster, $\{\mathcal{X}_k\} \in \hat{X}$. The distribution of the clustered space, on the other hand, is estimated as:

$$p_{\hat{X}}(\hat{x}) = \frac{|\mathcal{X}_k|}{|X|} \text{ for } k = 1, \dots, N \tag{13}$$

The joint distribution, $p_{X\hat{X}}(x, \hat{x})$, for each of the N clusters of the clustered space is estimated as:

$$p_{X\hat{X}}(x, \hat{x}) = p(x|\hat{x})p_{\hat{X}}(\hat{x}) = \frac{1}{|X|} \sum_{i=1}^{|\mathcal{X}_k|} \mathcal{N}(x - x_i, \delta_X^2) \text{ for } k = 1, \dots, N \tag{14}$$

Substituting (12), (13) and (14) in (11) provides us with the following approximation for the discrete quadratic mutual information (proof provided in Appendix):

$$\tilde{I}(X; \hat{X}) = \frac{1}{|X|^2} \left(\sum_{k=1}^N \sum_{i=1}^{|\mathcal{X}_k|} \sum_{j=1}^{|\mathcal{X}_k|} \mathcal{N}(x_i - x_j, 2\delta_X^2) - 2 \sum_{k=1}^N \frac{|\mathcal{X}_k|}{|X|} \sum_{i=1}^{|\mathcal{X}_k|} \sum_{j=1}^{|\mathcal{X}_k|} \mathcal{N}(x_i - x_j, 2\delta_X^2) + \left[\sum_{k=1}^N \frac{|\mathcal{X}_k|^2}{|X|^2} \right] \sum_{i=1}^{|\mathcal{X}_k|} \sum_{j=1}^{|\mathcal{X}_k|} \mathcal{N}(x_i - x_j, 2\delta_X^2) \right) \tag{15}$$

For simplification, here we define the between cluster distance among clusters \mathcal{X}_k and \mathcal{X}_l , as $\delta_{k,l} = \sum_{i=1}^{|\mathcal{X}_k|} \sum_{j=1}^{|\mathcal{X}_l|} \mathcal{N}(x_i - x_j, 2\delta_X^2)$, therefore, (15) can be represented as:

$$\tilde{I}(X; \hat{X}) = \frac{1}{|X|^2} \left(\sum_{k=1}^N \delta_{k,k} - 2 \sum_{k=1}^N \frac{|\mathcal{X}_k|}{|X|} \sum_{l=1}^N \delta_{k,l} + \kappa \sum_{k=1}^N \frac{|\mathcal{X}_k|^2}{|X|^2} \right) \tag{16}$$

where $\kappa = \sum_{i=1}^{|\mathcal{X}_k|} \sum_{j=1}^{|\mathcal{X}_k|} \mathcal{N}(x_i - x_j, 2\delta_X^2)$ is a constant.

2.4. Hierarchical Optimization

The proposed hierarchical algorithm, similar to most hierarchical clustering algorithms, operates in a bottom-up approach. In this approach clusters are merged together until one cluster is obtained, and then the whole process is evaluated to find the best number of clusters that fits the data [4]. Such clustering algorithms start by assuming each sample as an individual cluster, and therefore require $|X| - 1$ merging steps. To reduce the number of merging steps, hierarchical algorithms generally exploit an low-complexity initial clustering, such as k -means clustering, to generate \bar{N} clusters, far

beyond the expected number of clusters in the data, but still much smaller than the number of samples, $\bar{N} \ll |X|$ [23,24]. The initial clustering generates small spherical clusters, while significantly reducing the computational complexity of the hierarchical clustering.

Similarly, in the proposed hierarchical algorithms, clusters are merged; however, the criterion is to maximize the quadratic mutual information. Here we propose two approaches for merging, the agglomerative clustering and the split and merge clustering. In each hierarchy of the agglomerative clustering, two clusters are merged into one cluster to maximize the quadratic mutual information. In each hierarchy of the split and merge clustering, on the other hand, the cluster that has the worst effect on the quadratic mutual information is first eliminated, and then its samples are assigned to the remaining clusters in the clustered space. Following, these two approaches are explained in details.

2.4.1. Agglomerative Clustering

In this approach, we compute the changes in the quadratic mutual information after combining any pair of clusters to find the best two clusters for merging. We pick the pair that generates the largest increase in the quadratic mutual information. Since, at each hierarchy, clusters with the lowest distortion are generated, therefore, this approach can be used for optimizing the distortion-rate function in (5). Assuming that clusters \mathcal{X}_A and \mathcal{X}_B are merged to produce $\mathcal{X}_C = \mathcal{X}_A \cup \mathcal{X}_B$, the changes in the quadratic mutual information, $\Delta \tilde{I}_{A,B}$, can be estimated as:

$$\begin{aligned} \Delta \tilde{I}_{A,B}^{(t+1)} &= \tilde{I}^{(t+1)}(X; \hat{X}) - \tilde{I}^{(t)}(X; \hat{X}) \\ &= \frac{1}{|X|^2} \left(2\delta_{A,B} - \frac{2}{|X|} \left(|\mathcal{X}_A| \sum_{l=1}^N \delta_{B,l} + |\mathcal{X}_B| \sum_{l=1}^N \delta_{A,l} \right) + \frac{2\kappa|\mathcal{X}_A||\mathcal{X}_B|}{|X|^2} \right) \end{aligned} \quad (17)$$

where $\tilde{I}^{(t)}(X; \hat{X})$ is the quadratic mutual information at the step t . The closed form equation in (17) provides us with the best pair for merging without literally combining each pair and estimating the quadratic mutual information. Eventually, the maximum $\Delta \tilde{I}_{A,B}^{(t+1)}$ at each hierarchy determines the true number of clusters in the data. Table 1 introduces the pseudo code for the agglomerative clustering approach.

Table 1. Pseudo code for the agglomerative clustering.

1: Initial Clustering, $ \hat{X} = \bar{N}$
2: for $t = 1: \bar{N} - 1$ do
3: Estimate $\Delta \tilde{I}_{A,B}^{(t+1)}$ for all pairs
4: Merge clusters \mathcal{X}_A and \mathcal{X}_B , in which $\mathcal{X}_A, \mathcal{X}_B = \max_{A,B} \Delta \tilde{I}_{A,B}^{(t+1)}$
5: end for
6: Determine # of clusters

2.4.2. Split and Merge Clustering

Unlike the agglomerative clustering, this approach detects one cluster at each hierarchy for elimination. This cluster has the worst effect on the quadratic mutual information, meaning that out of all clusters, this is the cluster to be eliminated such that the mutual information is maximized.

Assuming cluster \mathcal{X}_A has the worst effect on the mutual information, the change in the quadratic mutual information, $\Delta \tilde{I}_A$, can be estimated as:

$$\begin{aligned} \Delta \tilde{I}_A^{(t+1)} &= \tilde{I}^{(t+1)}(X; \hat{X}) - \tilde{I}^{(t)}(X; \hat{X}) \\ &= \frac{1}{|X|^2} \left(\delta_{A,A} - 2 \left(\frac{|\mathcal{X}_A|}{|X|} \sum_{l=1}^N \delta_{A,l} + \sum_{k=1, k \neq A}^N \frac{|\mathcal{X}_k|}{|X|} \sum_{l=1}^N \delta_{k,A} \right) + \kappa \frac{|\mathcal{X}_A|^2}{|X|^2} \right) \end{aligned} \tag{18}$$

The samples of the worst cluster are then individually assigned to the remaining clusters of the clustered space based on the minimum Euclidean distance, in which the closest samples are assigned first. This process also proceeds until one cluster remains. Eventually, based on the maximum changes in the quadratic mutual information at different hierarchies, $\Delta \tilde{I}_A^{(t+1)}$, the true number of clusters is determined. Table 2 introduces the pseudo code for the split and merge clustering approach.

Table 2. Pseudo code for the split and merge clustering.

1: Initial Clustering, $ \hat{X} = \bar{N}$
2: for $t = 1: \bar{N} - 1$ do
3: Estimate $\Delta \tilde{I}_A^{(t+1)}$ for all pairs
4: Eliminate cluster \mathcal{X}_A , in which $\mathcal{X}_A = \max_A \Delta \tilde{I}_A^{(t+1)}$
5: for $i = 1: \mathcal{X}_A $ do
6: Assign sample $x_i = \min_{i,k} d(x_i, \hat{x} \in \mathcal{X}_k)$ to cluster \mathcal{X}_k
7: end for
8: end for
9: Determine # of clusters

Comparing the two proposed hierarchical algorithms, the split and merge clustering has the advantage of being unbiased to the initial clustering, since the eliminated cluster in each hierarchy is entirely re-clustered. However, the computational complexity of the split and merge algorithm is in the order of $O(\bar{N} \times |X|^2)$ and higher than the agglomerative clustering, that is in the order of $O(\bar{N}^2 \times |X|)$. The split and merge clustering also has the advantage of being less sensitive to the variance selection for the Gaussian kernels, since re-clustering is performed based on the minimum Euclidean distance.

Both proposed hierarchical approaches are unsupervised clustering algorithms; therefore require finding the true number of clusters. Determining the number of clusters is challenging, especially when no prior information is given about the data. In the proposed hierarchical clustering, we have access to the changes of the quadratic mutual information from the hierarchies. The true number of clusters is determined when the mutual information is maximized or when a dramatic change in the rate is observed.

Another parameter to be set for the proposed hierarchical clustering is the variance of the Gaussian kernels for the Parzen window estimator. Different variances detect different structures of data. Although there are no theoretical guidelines for choosing this variance, but some statistical methods can be used. For example, an approximation can be obtained for the variance in different dimensions, $\delta^2 = (1.06 \times \sum_{i=1}^d \delta_{ii}) / d \sqrt{|X|}$ where δ_{ii} is the diagonal element of the covariance matrix of the data [25]. We can also set the variance proportional to the minimum variance observed in each dimension, $\delta^2 = (1.06 \times \min\{\delta_{11}, \dots, \delta_{dd}\}) / \sqrt{|X|}$ [26].

3. Experimental

The performance of the proposed hierarchical clustering was evaluated by clustering both artificial and real data. The distribution patterns of samples in the artificial data were designed such that they incorporate clusters with different shapes and sizes. The nonlinear boundaries between clusters in these data make it impossible for linear clustering algorithms, such as k -means clustering, to detect the true clusters. In Figure 3, the clustering performance of both the agglomerative and split and merge clustering algorithms are demonstrated, in which the k -means clustering was used to produce the initial clusters. Figures 3a and 3b show data containing two bean-shape clusters, with a total of 796 samples. Figures 3c and 3d show data containing three concentric circle-shape clusters, connected by some random samples, totally containing 580 samples. Besides successful clustering by both of the two hierarchical approaches, slight differences in the clustering outcome are mainly due to the arbitrary nature of the initial clustering.

The clustering performance on real data, namely the Iris data and the Wine data, was used to make direct comparison with clustering algorithms reported in the literature. Using these data has the benefit of knowing the actual clusters, and can be used for evaluating clustering algorithms. The Iris data is one of the earliest test benchmarks for clustering that contains three clusters, each with 50 samples [27]. Every sample includes four features collected from an iris flower. The split and merge clustering generates six errors in the clustering result, compared to the actual labels, while the agglomerative clustering generates 10 errors. This is while an unsupervised perceptron network generates 19 errors, the superparamagnetic clustering misses 25 samples [28], the information based clustering generates 14 errors [15] and the information forced clustering generates 15 errors [29].

The Wine data demonstrates the chemical analysis of wines derived from different cultivars, and therefore resembles different clusters. This data was developed to use chemical analysis for determining the origin of wines [30]. The Wine data contains three clusters with 178 samples, each with 13 features that represent different constituents of the wine. The split and merge clustering generates 15 errors, which is far less than the 56 errors produced by fuzzy clustering [3].

We also evaluated the performance of the proposed hierarchical algorithms in detecting the true number of clusters, and compared it with other methods reported in the literature. An artificial data set with nine Gaussian distributed clusters, arranged in three groups of three clusters, is used for this purpose. The variance of each cluster was increased such that the boundaries between the clusters in each group would start fading, and the data seemingly would have three clusters instead of nine [31]. Figures 4a, 4c, and 4e show the data for distribution variances 0.02, 0.04 and 0.06, respectively. By implementing the split and merge clustering, we demonstrate the mean quadratic mutual information estimated at each hierarchy in Figures 4b, 4d, and 4f, in which the errorbar shows the standard deviation for repeating the clustering 10-fold, each originating from a different initial clustering. As the variance increases for the Gaussian distributed clusters, the peak of mutual information deliberately shifts from nine clusters to three clusters. The advantage of the proposed clustering in detecting the true number of clusters is best illustrated in Figures 4d and 4f, where it progressively indicates the confusion between choosing three or nine clusters for this data. This confusion is represented as a local maximum at nine clusters with a global maximum at three clusters. Most algorithms mistakenly detect three clusters instead of nine clusters, especially for the data in Figure 4e, such as an information-theoretic approach

proposed for finding the number of clusters [31] and clustering based on Rényi’s entropy, which uses the variations of between cluster entropy for detecting the true number of clusters [32].

Figure 3. The clustering results obtained for the data with two bean-shaped clusters by (a) agglomerative clustering and (b) split and merge clustering. The clustering results obtained for the data with three concentric circle-shaped clusters by (c) agglomerative clustering and (d) split and merge clustering.

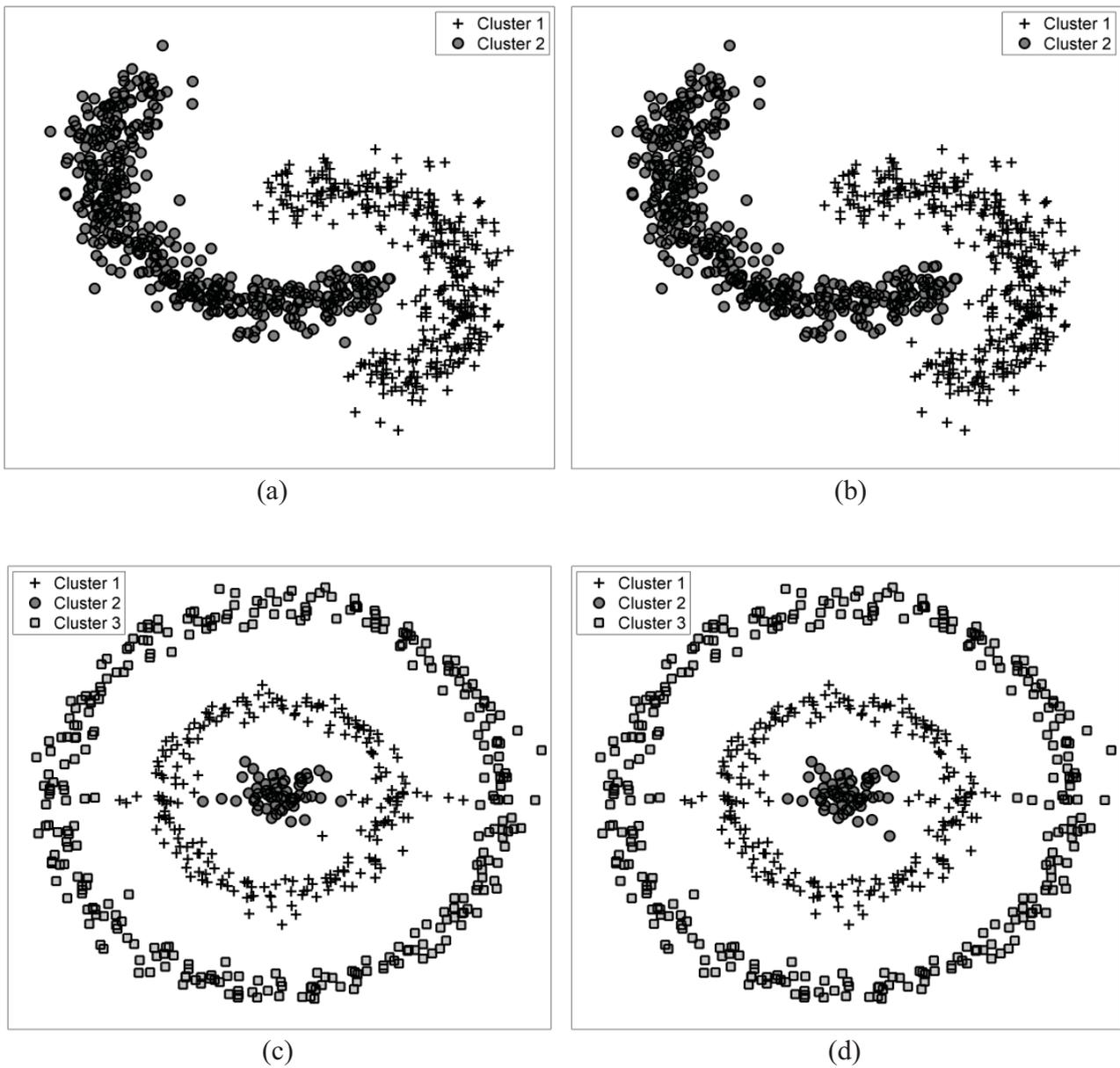
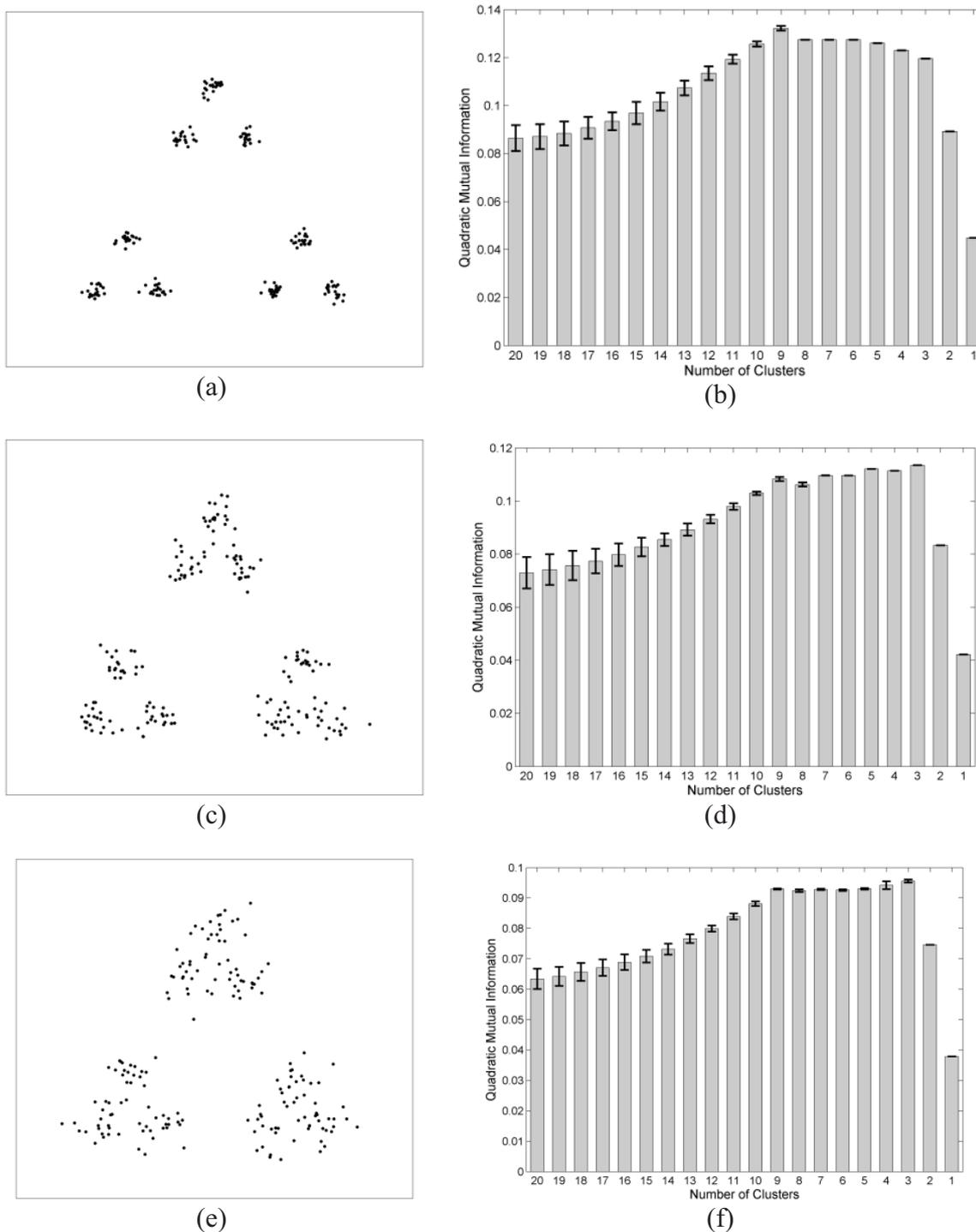


Figure 4. Data with 9 Gaussian distributed clusters arranged in three groups with variances (a) 0.02, (c) 0.04, and (e) 0.06. The mean quadratic mutual information and its standard deviation measured at each step of the split and merge clustering for ten trials is demonstrated. A clear global maximum at nine clusters is observed for (b). Subplots (d) and (f) show a local maximum at nine clusters and a global maximum at three, demonstrating the confusion in selecting nine clusters out of three group of clusters when the variance is relatively high.



4. Conclusions

In this paper, mutual information has been used as an information theoretic proximity measure for clustering. Although information theoretic measures facilitate extracting data structures further than the second order statistics, but are challenged by the practical difficulties in estimating the distribution of samples in each cluster. We use the non-parametric Parzen window estimator with Gaussian kernels to estimate this distribution, where it has been shown to simplify the quadratic approximation of the mutual information into a sum of pairwise distances. Since the quadratic mutual information can be updated iteratively for newly generated clusters, it is an appropriate proximity measure for hierarchical clustering algorithms.

Two hierarchical approaches are proposed for maximizing the quadratic mutual information between the samples of the input space and the clusters, namely the agglomerative and the split and merge clustering. We demonstrated how maximizing the mutual information is analogous to optimizing the distortion-rate function that achieves the minimum distortion among samples of each clusters for a given rate, and therefore provides the best clustering result. Beginning from a preliminary set of clusters generated by the initial clustering, the agglomerative clustering finds the best pair of clusters to be merged at each step of the hierarchy, while the split and merge clustering eliminates the cluster with the worst effect on the mutual information, and reassigns its samples to the remaining clusters. Finally, the true number of clusters in the data is determined based on the rate of changes in the quadratic mutual information from different hierarchies. Although the split and merge clustering is computationally more expensive than the agglomerative clustering, it benefits from being less sensitive to the initial clustering and the selection of the variance for the Gaussian kernels.

Experimental results on both artificial and real datasets have illustrated the promising performance of our proposed algorithms, both in finding the nonlinear boundaries between complex-shape clusters and determining the true number of clusters in the data. While conventional clustering algorithms typically fail in detecting the true data structures of the introduced artificial datasets, the proposed algorithms were able to detect clusters with different shapes and sizes. The superiority of the proposed algorithms for real applications is demonstrated by their clustering performance on the Iris and Wine datasets in comparison with some clustering algorithms reported in the literature.

References

1. Mahalanobis, P.C. On the generalized distance in statistics. *Proc. Natl. Inst. Sci. India* **1936**, *12*, 49–55.
2. Hartigan, J.A.; Wong, M.A. A k-means clustering algorithm. *JR Stat. Soc. Ser. C-Appl. Stat.* **1979**, *28*, 100–108.
3. Bezdek, J.C.; Ehrlich, R. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203.
4. Johnson, S.C. Hierarchical clustering schemes. *Psychometrika* **1967**, *32*, 241–254.
5. Jain, A.K.; Mao, J.; Mohiuddin, K.M. Artificial neural networks: A tutorial. *Computer* **1996**, *29*, 31–44.
6. Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines: and Other Kernel-based Learning Methods*; Cambridge University Press: Cambridge, UK, **2000**.

7. Davis, J.V.; Kulis, B.; Jain, P.; Sra, S.; Dhillon, I.S. Information-theoretic metric learning. In Proceedings of the 24th international conference on Machine learning (ICML '07), New York, NY, USA, 2007; pp. 209–216.
8. Chaudhuri, K.; McGregor, A. Finding metric structure in information theoretic clustering. In Conference on Learning Theory, COLT, Helsinki, Finland, July 2008.
9. Roweis, S.; Ghahramani, Z. A unifying review of linear Gaussian models. *Neural Comput.* **1999**, *11*, 305–345.
10. Parzen, E. On estimation of a probability density function and mode. *Ann. Math. Statist.* **1962**, *33*, 1065–1076.
11. Kraskov, A.; Grassberger, P. MIC: mutual information based hierarchical clustering. In *Information Theory Statistical Learning*; Springer: New York, NY, USA, 2008, pp. 101–123.
12. Zhou, X.; Wang, X.; Dougherty, E.R.; Russ, D.; Suh, E. Gene clustering based on clusterwise mutual information. *J. Comput. Biol.* **2004**, *11*, 147–161.
13. Rényi, V. On measures of information and entropy. Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability, Berkeley: University of California Press, California, USA, 1961; pp. 547–561.
14. Principe, J.C.; Xu, D.; Fisher, J. Information theoretic learning. In *Unsupervised Adaptive Filtering*, Haykin, S., Ed.; Wiley: New York, NY, USA, 2000; pp. 265–319.
15. Gokcay, E.; Principe, J.C. Information theoretic clustering. *IEEE Trans. Patt. Anal. Mach. Int.* **2002**, *24*, 158–171.
16. Torkkola, K. Feature extraction by non parametric mutual information maximization. *J. Mach. Learn. Res.* **2003**, *3*, 1438.
17. Sullivan, G.J.; Wiegand, T. Rate-distortion optimization for video compression. *Signal Process. Mag. IEEE* **2002**, *15*, 74–90.
18. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423.
19. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*; 2nd ed.; Wiley: New York, NY, USA, 2006.
20. Kapur, J.N. *Measures of Information and Their Applications*; Wiley: New Delhi, India, 1994.
21. Vincent, P.; Bengio, Y. Manifold parzen windows. *Adv. Neural Inf. Process. Syst.* **2003**, *15*, 849–856.
22. Davis, J.V.; Dhillon, I. Differential entropic clustering of multivariate gaussians. *Adv. Neural Inf. Process. Syst.* **2007**, *19*, 337.
23. Karypis, G.; Han, E.H.; Kumar, V. Chameleon: Hierarchical clustering using dynamic modeling. *Computer* **1999**, *32*, 68–75.
24. Guha, S.; Rastogi, R.; Shim, K. Cure: An efficient clustering algorithm for large databases. *Inf. Syst.* **2001**, *26*, 35–58.
25. Duda, R.O.; Hart, P.E. Pattern classification and scene analysis. Wiley, New York, NY, USA, 1973.
26. Silverman, B.W. *Density Estimation for Statistics and Data Analysis*; Chapman & Hall/CRC: London, England, UK, 1998.
27. Anderson, E. The irises of the Gaspé Peninsula. *Bull. Am. IRIS Soc.* **1935**, *59*, 2–5.
28. Blatt, M.; Wiseman, S.; Domany, E. Superparamagnetic clustering of data. *Phys. Rev. Lett.* **1996**, *76*, 3251–3254.

29. Jenssen, R.; Erdogmus, D.; Ii, K.; Principe, J.; Eltoft, T. Information force clustering using directed trees. *Lect. Note. Comput. Sci.* **2003**, 2683, 68–72.
30. Aeberhard, S.; Coomans, D.; De Vel, O. Comparison of classifiers in high dimensional settings. *Tech. Rep.* **1992**, 92–102.
31. Sugar, C.A.; James, G.M. Finding the number of clusters in a dataset. *J. Am. Statist. Assn.* **2003**, 98, 750–763.
32. Jenssen, R.; Hild, K.E.; Erdogmus, D.; Principe, J.C.; Eltoft, T. Clustering using Renyi’s entropy. *Neural Networks* **2003**, 1, 523–528.

Appendix

Here we provide a proof for (15). The proof for each term, out of the three quadratic terms in (11), is presented separately. Starting from the first term we have:

$$\begin{aligned}
 & \sum_{x \in X} \sum_{\hat{x} \in \hat{X}} p_{X\hat{X}}^2(x, \hat{x}) \\
 &= \sum_{x \in X} \sum_{\hat{x} \in \hat{X}} \frac{1}{|X|^2} \left[\sum_{i=1}^{|\mathcal{X}_k|} \mathcal{N}(x - x_i, \delta_{\hat{X}}^2) \right] \left[\sum_{i=1}^{|\mathcal{X}_k|} \mathcal{N}(x - x_i, \delta_{\hat{X}}^2) \right] \\
 &= \frac{1}{|X|^2} \sum_{\hat{x} \in \hat{X}} \sum_{x \in X} \left[\sum_{i=1}^{|\mathcal{X}_k|} \mathcal{N}(x - x_i, \delta_{\hat{X}}^2) \right] \left[\sum_{i=1}^{|\mathcal{X}_k|} \mathcal{N}(x - x_i, \delta_{\hat{X}}^2) \right] \\
 &= \frac{1}{|X|^2} \sum_{\hat{x} \in \hat{X}} \sum_{i=1}^{|\mathcal{X}_k|} \sum_{j=1}^{|\mathcal{X}_k|} \mathcal{N}(x_i - x_j, 2\delta_{\hat{X}}^2) \\
 &= \frac{1}{|X|^2} \sum_{k=1}^N \sum_{i=1}^{|\mathcal{X}_k|} \sum_{j=1}^{|\mathcal{X}_k|} \mathcal{N}(x_i - x_j, 2\delta_{\hat{X}}^2)
 \end{aligned}$$

Similarly, for the second term we have:

$$\begin{aligned}
 & \sum_{x \in X} \sum_{\hat{x} \in \hat{X}} p_{X\hat{X}}(x, \hat{x}) p_X(x) p_{\hat{X}}(\hat{x}) \\
 &= \sum_{x \in X} \sum_{\hat{x} \in \hat{X}} \frac{1}{|X|^2} \left[\sum_{i=1}^{|\mathcal{X}_k|} \mathcal{N}(x - x_i, \delta_{\hat{X}}^2) \right] \left[\sum_{i=1}^{|\mathcal{X}_k|} \mathcal{N}(x - x_i, \delta_{\hat{X}}^2) \right] \frac{|\mathcal{X}_k|}{|X|} \\
 &= \frac{1}{|X|^2} \sum_{\hat{x} \in \hat{X}} \frac{|\mathcal{X}_k|}{|X|} \sum_{x \in X} \left[\sum_{i=1}^{|\mathcal{X}_k|} \mathcal{N}(x - x_i, \delta_{\hat{X}}^2) \right] \left[\sum_{i=1}^{|\mathcal{X}_k|} \mathcal{N}(x - x_i, \delta_{\hat{X}}^2) \right] \\
 &= \frac{1}{|X|^2} \sum_{\hat{x} \in \hat{X}} \frac{|\mathcal{X}_k|}{|X|} \sum_{i=1}^{|\mathcal{X}_k|} \sum_{j=1}^{|\mathcal{X}_k|} \mathcal{N}(x_i - x_j, 2\delta_{\hat{X}}^2) \\
 &= \frac{1}{|X|^2} \sum_{k=1}^N \frac{|\mathcal{X}_k|}{|X|} \sum_{i=1}^{|\mathcal{X}_k|} \sum_{j=1}^{|\mathcal{X}_k|} \mathcal{N}(x_i - x_j, 2\delta_{\hat{X}}^2)
 \end{aligned}$$

Finally, for the third term we have:

$$\begin{aligned}
 & \sum_{x \in X} \sum_{\hat{x} \in \hat{X}} p_X^2(x) p_{\hat{X}}^2(\hat{x}) \\
 &= \left[\sum_{\hat{x} \in \hat{X}} p_{\hat{X}}^2(\hat{x}) \right] \left[\sum_{x \in X} p_X^2(x) \right] \\
 &= \left[\sum_{k=1}^N \left(\frac{|\mathcal{X}_k|}{|X|} \right)^2 \right] \times \frac{1}{|X|^2} \sum_{i=1}^{|X|} \sum_{j=1}^{|X|} \mathcal{N}(x_i - x_j, 2\delta_X^2)
 \end{aligned}$$

© 2011 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).