A HIERARCHICAL CLUSTERING BASED ON MUTUAL INFORMATION MAXIMIZATION

M. Aghagolzadeh[a]*, H. Soltanian-Zadeh*[a, c]*, B. Araabi*[a]*, A. Aghagolzadeh*[b]

[a] Control and Intelligent Processing Center of Excellence, Department of Electrical and Computer Engineering, University of Tehran, Tehran 14395-515, Iran
[b] Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, 51664, Iran
[c] Radiology Image Analysis Lab., Henry Ford Health System, Detroit, MI 48202, USA

ABSTRACT

Mutual information has been used in many clustering algorithms for measuring general dependencies between random data variables, but its difficulties in computing for small size datasets has limited its efficiency for clustering in many applications. A novel clustering method is proposed which estimates mutual information based on information potential computed pair-wise between data points and without any prior assumptions about cluster density function. The proposed algorithm increases the mutual information in each step in an agglomerative hierarchy scheme. We have shown experimentally that maximizing mutual information between data points and their class labels will lead to an efficient clustering. Experiments done on a variety of artificial and real datasets show the superiority of this algorithm, besides its low computational complexity, in comparison to other information based clustering methods and also some ordinary clustering algorithms.

Index Terms— agglomerative hierarchical clustering, information potential, mutual information (*MI*), Renyi's entropy

1. INTRODUCTION

Clustering is an unsupervised classification of a dataset into its natural groups, so that the data in a labeled group have the highest similarity among themselves and the highest dissimilarity with the data in other groups. After a successful clustering, the within similarity between data pairs of a single cluster is maximized and the between similarity of data pairs assigned to different clusters is minimized. The most important part of any clustering method is the utilized similarity measure. Different clustering algorithms are associated with a special definition of distance or similarity [14]. Most clustering algorithms, because of using the intra-cluster variance or Euclidean distances, can only detect data structure limited to the second order statistics. A useful tool to extract complex data structures for clustering is information theory that is able to extract data structures further than the second order statistics. However, practical difficulties in approximating data's probability density function has limited its usage in clustering methods and increased the computational complexity. Density functions can be estimated by parametric or non-parametric methods [3]. Nonparametric methods are not limited to a special form or model and are more flexible than parametric methods but are computationally more expensive. The Parzen window estimator is a widely used nonparametric method for density estimation [9]. It was shown by Principe et al [10] that using the Parzen's window estimator to compute the Renyi's definition of entropy will lead to a simplified pair-wise calculation between data points in a dataset.

This paper addresses an information theoretic method which utilizes *MI* as a proximity measure. *MI* extracts data structures related to higher order statistics, further than only the second order statistics. *MI* has been utilized in many methods for obtaining more effective clustering techniques. In a hierarchical clustering method presented by Kraskov et al [8], *MI* is computed based on the grouping property for the next steps of the Mutual Information Clustering (*MIC*) algorithm. Estimating *MI* by this method, in general, is not easy especially in small size datasets with a few number of data points which causes to suboptimal results. In another method proposed by Zhou et al [17], a clustering strategy based on minimizing *MI* is applied among gene clusters. A simulated annealing algorithm is employed to optimize a cost function based on *MI* and minimize it. Because of the small size of data points and the difficulty in estimating the density function, they employed a bootstrap technique to achieve more accurate estimates of *MI* and increase the efficiency of clustering.

Principe et al [10] proposed a quadratic distance measure between probability density functions and estimated the density functions by Parzen window estimator. They showed that the quadratic divergence measure can be simplified in terms of Gaussian functions, computed based on the difference between data pairs. These kinds of quantities in analogy to physical particles are known as information potential. Torkkola [15] has presented a method for learning discriminant feature transforms using *MI* estimated between class labels and transformed features as a criterion.

In order to estimate *MI* in the proposed algorithm, the quadratic divergence measure is utilized. In any step of this algorithm, two clusters are combined to produce one new cluster. Combining or dividing clusters, and in general changing cluster labels, will change the *MI* for a dataset. Therefore, the two combining clusters are chosen to maximally increase *MI* in each step. Once establishing the algorithm, the proximity matrix of a desired dataset is computed by estimating the similarity between all data pairs with a quantity called information datum. To reduce the computational complexity, the proposed algorithm starts from a finite number of initial clusters, which are extracted at the beginning by the initial clustering. The appropriate number of clusters in a dataset can be determined by detecting the number of impulsive jumps in a dendrogram.

Experiments show that maximizing the *MI*, computed from the quadratic divergence measure between data points and their cluster labels, will lead to an appropriate clustering. Since *MI* for newly generated clusters at each step can be computed recursively from the *MI* of the combined clusters, this algorithm is computationally efficient. The main advantage of the proposed algorithm is its ability to detect nonlinear complex structures in a dataset. The detectable clusters by this method are not limited to a special prototype or shape. Experiments done on both artificial and real datasets show the superiority of this algorithm.

2. ESTIMATING MUTUAL INFORMATION

For presently developed information theoretic clustering methods based on *MI*, the Shannon's definition of information theory has been used for estimating *MI*, which can also be represented as the Kullback-Leibler divergence measure between *p(x, y)* and *p(x)p(y)*. However, computing *MI* by this method will encounter practical difficulties especially for small size datasets. This problem has greatly affected the efficiency of clustering results by these *MI* clustering algorithms. Since the purpose of clustering methods is not calculating *MI* in a dataset, but rather it is just for recognizing a distribution among data points which minimizes or maximizes a quantity of divergence, other criterions for divergence can be utilized [7]. Assume that a desired dataset has *N* samples, shown by a discrete random variable *X* in the R^N space. Each of these data points are assigned to one of the *n* clusters; and each sample x_i has a corresponding cluster label c_i. Therefore, *MI* can be defined based on the quadratic divergence measure.

$$MI(C,x) = \sum_C \sum_X \left(P(C,x) - P(C)P(x)\right)^2$$
$$= \sum_C \sum_X P(C,x)^2 + \sum_C \sum_X P(C)^2 P(x)^2 - 2\sum_C \sum_X P(C,x)P(C)P(x) \tag{1}$$

In the above equation, *P(C, x)* is the joint probability between cluster labels and data points and *P(x)* is the sample's distribution function. Probability of cluster labels, *P(C)*, is simply computed by dividing the number of samples in each cluster by the total number of samples in a desired dataset [15]. *P(C, x)* and *P(x)* can be estimated nonparametrically by the Parzen's window estimator, resulting the following equations.

$$P(x) = \frac{1}{N}\sum_{i=1}^{N} G\left(x - x_i, \sigma^2\right) \tag{2}$$

$$P(C_p, x) = \frac{1}{N}\sum_{k=1}^{N_p} G\left(x - x_{p,k}, \sigma^2\right) \tag{3}$$

Where N_p is the number of samples in cluster C_p and $x_{p,\,k}$ is a sample belonging to this cluster. $P(C_p, x)$ is computed for all *n* clusters. The Gaussian function, G is defined as

$$G\left(x - x_i, \sigma^2\right) = \frac{1}{(2\pi\sigma)^{\frac{d}{2}}} \exp\left(-\frac{\left\|x - x_i\right\|^2}{2\sigma^2}\right) \tag{4}$$

By placing (2) and (3) in (1), *MI* can be written in terms of Gaussian functions.

$$MI(C,x) = \sum_{p=1}^{n}\sum_X \left[\frac{1}{N^2}\sum_{i=1}^{N_p}\sum_{j=1}^{N_p} G\left(x - x_{p,i}, \sigma^2\right)G\left(x - x_{p,j}, \sigma^2\right)\right]$$
$$+ \left(\sum_{p=1}^{n}\left(\frac{N_p}{N}\right)^2\right)\left(\frac{1}{N^2}\sum_X\sum_{i=1}^{N}\sum_{j=1}^{N} G\left(x - x_i, \sigma^2\right)G\left(x - x_j, \sigma^2\right)\right)$$
$$- 2\sum_{p=1}^{n}\frac{N_p}{N}\left[\frac{1}{N^2}\sum_X\sum_{i=1}^{N}\sum_{j=1}^{N_p} G\left(x - x_i, \sigma^2\right)G\left(x - x_{p,j}, \sigma^2\right)\right] \tag{5}$$

Since the convolution of two Gaussian functions is also a Gaussian function, (5) can be simplified into the following form.

$$MI(C,x) = \sum_{p=1}^{n}\left[\frac{1}{N^2}\sum_{i=1}^{N_p}\sum_{j=1}^{N_p} G\left(x_{p,i} - x_{p,j}, 2\sigma^2\right)\right]$$
$$+ \left(\sum_{p=1}^{n}\left(\frac{N_p}{N}\right)^2\right)\left(\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G\left(x_i - x_j, 2\sigma^2\right)\right)$$
$$- 2\sum_{p=1}^{n}\frac{N_p}{N^3}\left[\sum_{i=1}^{N}\sum_{j=1}^{N_p} G\left(x_i - x_{p,j}, 2\sigma^2\right)\right] \tag{6}$$

The information between x_i and x_j is defined as an information datum, $I_{i,\,j} = G\ (x_i\text{-}x_j,\ 2\sigma^2)$. The Total Information Potential (*TIP*) is the summation of all information datums in a dataset. The summation of all information datums limited to a unique cluster is known as the within information potential of that cluster, WIP_p. The sum of all these WIP_p is the Within Information Potential between samples in a dataset (*WIP*). By defining the information potential between cluster C_p and the whole dataset as

$$CIP_p = \sum_{i=1}^{N_p}\sum_{j=1}^{N} G\left(x_{p,i} - x_j, \sigma^2\right) \tag{7}$$

equation (6) can be shortened into (8) which is based on only WIP_p, CIP_p, *TIP* and the number of samples in each cluster.

$$MI(C,x) = WIP + TIP \times \left(\sum_{p=1}^{n}\left(\frac{N_p}{N}\right)^2\right) - \frac{2}{N^3}\sum_{p=1}^{n}\left(N_p \times CIP_p\right) \tag{8}$$

3. PROPOSED HIERARCHICAL CLUSTERING

Maximizing the computed *MI* between samples and their cluster labels will force cluster labels to be assigned so that there will be more similarity between data points in each cluster, which is the main idea of a clustering algorithm. But an important challenging question appears: how should the cluster labels alter to achieve this goal?

To maximize *MI*, an agglomerative hierarchical clustering algorithm is proposed. In an agglomerative hierarchical clustering, clusters at each step merge to produce new clusters and this process is done until one cluster remains. In the proposed algorithm, the two merging clusters are chosen so that *MI* is increased. To reach to the maximum *MI*, variations of *MI* caused by combining any two clusters are computed and the pair of clusters which cause the maximum variation are merged. However merging any two clusters in a dataset and computing the variation in *MI* seems computationally expensive, but we will show that these variations can simply be computed based on only *WIP* and *CIP* of those two clusters.

Suppose that cluster C_a is to be combined with cluster C_b to generate cluster C_p. Variations in *MI* caused by combining two clusters is simply computed by the following.

$$\Delta MI(C,x) = \left(WIP_p - WIP_a - WIP_b\right) + TIP \times \frac{N_p^2 - N_a^2 - N_b^2}{N^2}$$
$$- 2\frac{N_p \times CIP_p - N_a \times CIP_a - N_b \times CIP_b}{N^3} \tag{9}$$

Since

$$WIP_p = WIP_a + WIP_b + \frac{2}{N^2}\sum_{i=1}^{N_a}\sum_{j=1}^{N_b} G\left(x_i - x_j, \sigma^2\right) \tag{10}$$

And

$$CIP_p = CIP_a + CIP_b \tag{11}$$

Therefore, equation (9) can be simplified into the following form.

$$\Delta MI=\frac{2}{N^2}\sum_{i=1}^{N_a}\sum_{j=1}^{N_b}G\left(x_i-x_j,\sigma^2\right)+\frac{2N_aN_b}{N^2}TIP-2\frac{N_a\times CIP_b+N_b\times CIP_a}{N^3} \qquad (12)$$

Consequently, the pair of clusters which has the maximum increase in *MI* is chosen to be the best choice for merging at any step of the proposed clustering algorithm.

4. ESTIMATING NUMBER OF CLUSTERS

In many clustering applications, the actual number of clusters is unknown; therefore an important factor of any clustering algorithm is to determine the actual number of clusters or in other words, the clustering method that estimates the number of clusters more appropriately, will also generate a better clustering result. A better clustering is achieved when within cluster similarity, and between cluster dissimilarity are maximized together with the minimum number of clusters. Therefore different clustering algorithms depending on the utilized similarity and dissimilarity measurements give different number of clusters.

The proposed clustering algorithm gathers data points in a dataset to generate larger clusters and this is done until one single cluster is achieved. If no clustering is applied on the data points before applying the proposed algorithm, then any data point is assumed to be an individual cluster. Consequently, the final clustering by the proposed method will generate the best clustering with high quality and stability, but with a high execution time. On the other hand, if some initial clusters are produced by a suggested clustering method, then applying the proposed clustering algorithm will execute much less steps, but this might influence the clustering result and decrease the quality and stability. Therefore, there is a trade off between the computational complexity and the quality/stability.

The number of initial clusters depends on the dataset. The number of samples and structure of the dataset specify the number of initial clusters. When a dataset contains line and shell prototype clusters, the number of initial clusters is greater than when it contains only mass prototype clusters. Experiments show that by setting the number of initial clusters to *N/10*, the final clustering is not influenced with this fact that the computational complexity is low. K-nearest neighbor clustering is applied on a desired dataset to limit the initial number of clusters [3]. For this reason *P* data points are selected randomly to be the initial seeds and then any data point is allocated to the nearest seed based on Euclidean distance. This action has a low affect on the clustering result.

To determine the final clustering a hierarchical structure is supposed for the propose algorithm. The main advantage of a hierarchical clustering in comparison to partitional clustering is that a dendrogram can be drawn to find the appropriate number of clusters in a dataset. Unlike ordinary hierarchical algorithms which are based on Euclidean distance, the length in the dendrogram in the proposed clustering, also known as the lifetime, is the inverse of variations in *MI*. The best place to stop the hierarchical algorithm before ending to one cluster is when the increasing ascension of *MI* decreases. Therefore, the inverse of *MI* variations (*1/*Δ*MI*) is utilized as a tool for detecting the final clustering. The final number of clusters is set to the number of sudden jumps in the dendrogram. Figure 1.a shows a dataset with four distinctive nonlinear mass clusters. The dendrogram for this dataset is shown in the Figure 1.b. The dendrogram contains four jumps, therefore, the final number of clusters is set to four clusters. It must be mentioned that any jump after a sudden detected jump is not considered for final clustering.

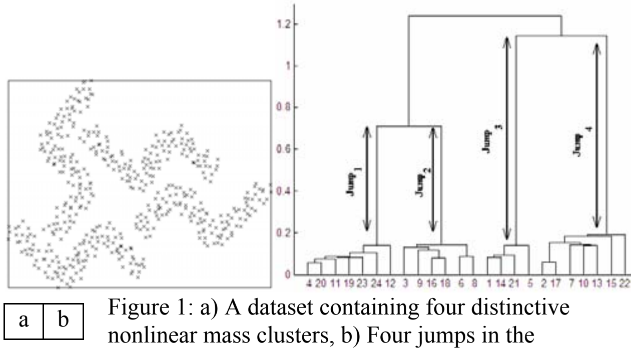


Figure 1: a) A dataset containing four distinctive nonlinear mass clusters, b) Four jumps in the dendrogram defines four clusters as the final number of clusters.

5. EXPERIMENT RESULTS

To evaluate the efficiency of the proposed clustering algorithm, experiments are done on both the artificial datasets and the Iris dataset as a real dataset. Artificial datasets are produced manually to represent real data. In designing an artificial dataset, complex structures are used which are regularly more complicated than real datasets. The clustering methods that are able to cluster these complicated artificial datasets are also able to cluster real datasets. The artificial datasets are designed so that simple clustering algorithms like C-means, fuzzy clustering and linear clustering algorithms are not able to detect the actual clusters. Most of these artificial datasets are used by recently clustering algorithms to evaluate their clustering methods. The proposed algorithm is also compared with some other clustering algorithms by clustering a real dataset (Iris dataset).

Before applying the proposed clustering on any dataset, the kernel size must be assigned. The main problem of most of the algorithms which uses Gaussian functions in the Parzen window estimator is that there are no theoretical guidelines to choose a kernel size. These methods often have high sensitivity to variations of *σ*, so that for values bigger or smaller than the correct kernel size, the clustering result will change generally and fail. By choosing a small kernel size, attention is given to the close data points which produce clusters covering only nearby data points. In this manner clusters have mass prototype or are compressed, without containing any outliers or correlated members. By choosing a larger kernel size, attention is also given to the far data points but it can also make the whole clustering unstable. This shows the complexity of choosing the kernel size and its dependence on the dataset structure.

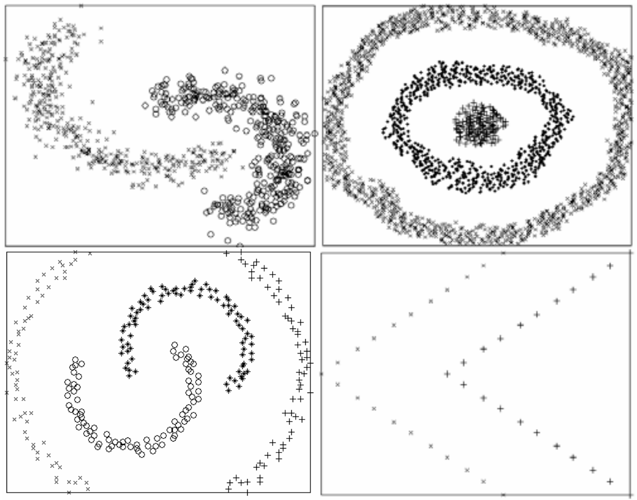
To choose the kernel size automatically, methods have been developed which estimate kernel size from the properties of the dataset and its data point's distribution. A simple method for choosing the kernel size is utilized in the proposed algorithm [13].

$$\sigma=\min\left\{S_{Dimension\_x},\ldots,S_{Dimension\_z}\right\}\times 1.06\times N^{-0.2} \qquad (13)$$

where *S* is the diagonal element of the dataset's covariance matrix, in the direction of one of the feature vectors. Using the manifold Parzen window is another suitable method for nonlinear data structures which chooses the kernel size for each data point locally [16]. Despite the mentioned methods for selecting the kernel size, often the actual kernel size differs with the kernel sizes presented by the above methods and many algorithms select the kernel size manually.

5.1. Artificial Datasets

Figures 2.a to 2.d show datasets with centralized clusters, which have complex nonlinear structures. To detect the clusters correctly, information must be extracted further than the second order statistics. Figure 2.d shows a dataset with extremely low number of samples. In contrast to many *MI* based clustering algorithms, the proposed algorithm is designed to cluster small size datasets too.



a	b
c	d

Figure 2: clustering results by the proposed algorithm

5.2. Iris Dataset

The Iris dataset is one of the oldest databases collected by Anderson [1] and used by Fisher [4] to compare clustering algorithms. It contains 150 data points in 3 clusters, 50 data points in each cluster. Each cluster represents one kind of Iris flower and each data point has four features.

Table 1: number of misclassifications in clustering the Iris dataset

Clustering method	Number of misclassifications
Unsupervised Perceptron	19 errors
Gokcay and Principe [5]	14 errors
Jenssen et al [6]	15 errors
Proposed Clustering	10 errors

The proposed algorithm, when applied on the Iris dataset, produces different number of errors in consecutive repetitions. The minimum produced mistakes are equal to 4 errors while the maximum produced mistakes are 18 errors. This shows the high dependence to the initial clustering and its random-seed selecting procedure. By eliminating the initial clustering process at the beginning of the proposed algorithm and assuming each data point as an individual cluster, 10 errors are produced, that is better than recently developed information theoretic clustering methods, which have been tested on the Iris dataset (Table 1).

6. CONCLUSION

In this paper, a novel hierarchical clustering based on *MI* is proposed. *MI* is defined in terms of information potential by employing the quadratic divergence measure. Maximizing the *MI* between data points and their cluster labels results in an efficient clustering. For this reason, an agglomerative hierarchical clustering is designed which increases *MI* in each step. Variations of *MI* caused by combining any pair of clusters are computed and then the maximum amount is chosen to define the two clusters to be merged in that step. For finding the best clustering, a dendrogram based on the inverse of *MI* variations is drawn. The number of sudden jumps determines the final number of clusters in a dataset. The computational complexity of this algorithm for computing the proximity matrix, like other hierarchical clustering algorithms, is in the order of $O(N^2)$. Since the proposed algorithm begins from P initial clusters, the computational complexity of the remainder of the algorithm is in the order of $O(P^2)$. Since $P<<N$, the proposed algorithm can be considered as a very fast hierarchical algorithm.

11. REFERENCES

[1] E. Anderson, “The Irises of the Gaspe peninsula,” *Bulletin of the American Iris Society*, vol. 59, pp. 2-5, 1935.

[2] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, *Plenum Press*, New York, 1981.

[3] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification and Scene Analysis, *John Wiley & sons*, 2001.

[4] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annuals of Eugenics*, vol. 7, pp. 179-188, 1936.

[5] E. Gokcay, and J. C. Principe, “Information Theoretic Clustering,” *IEEE Transaction on PAMI*, Vol. 24, No. 2, February pp. 158-171, 2002.

[6] R. Jenssen, D. Erdogmus, K. E. Hild, J. C. Principe and T. Eltoft, “Information force clustering using directed trees,” *EMMCVPR*, pp. 68-82, 2003.

[7] J. N. Kapur, Measures of information and their applications, *Wiley*, New Delhi, India, 1994.

[8] A. Kraskov, H. Stogbauer, R. G. Andrzejak, and P. Grassberger, “Hierarchical Clustering Based on Mutual Information,” *http://arxiv.org/abs/q-bio/0311039*, 2003.

[9] E. Parzen, “On the estimation of probability density function and the mode, *the Annals of Mathematical Statistics*, vol. 33, pp. 1065, 1962.

[10] J. C. Principe, D. Xu, and J. W. Fisher III, “Information theoretic learning,” in: S. Haykin (Ed.), Unsupervised Adaptive Filtering: Blind Source Separation, *John Wiley & Sons Inc*., pp. 265-319, 2000.

[11] A. Renyi, “On Measures of Entropy and Information,” *in Proceedings of the Fourth Berkeley Sympodium on Mathematics of Statistics and Probability*, vol. 1, pp. 547-561, 1961.

[12] C. E. Shannon, “A Mathematical Theory of Communication,” *Bell System Technical Journal*, Vol. 27, July pp 379-423, October pp. 623-656, 1948.

[13] B. W. Silverman, Density estimation for statistics and data analysis, *Chapman and Hall*, 1986.

[14] S. Theodoridis and K. Koutroumbas, Pattern Recognition, *Academic Press*, 1999.

[15] K. Torkkola, “Feature Extraction by Non-Parametric Mutual Information Maximization,” *Journal of Machine Learning Research*, vol. 3, pp 1415-1438, 2003.

[16] P. Vincent and Y. Bengio, “Manifold Parzen windows,” *Advances in Neural Information Processing Systems*, vol. 12, MIT Press, pp 825-832, 2003.

[17] X. Zhou, X. Wang, E. R. Dougherty, D. Russ, and E. Suh, “Gene Clustering Based on Clusterwide Mutual Information,” *Computational Biology*, vol. 11, no. 1, pp. 147-161, 2004.