Minimum Entropy Clustering and Applications to Gene Expression Analysis

Haifeng Li[†], Keshu Zhang[‡], and Tao Jiang[†]

[†] Department of Computer Science and Engineering
University of California, Riverside, CA 92521
{hli, jiang}@cs.ucr.edu

[‡] Human Interaction Research Lab
Motorola, Inc., Tempe, AZ 85282
keshu.zhang@motorola.com

Abstract

Clustering is a common methodology for analyzing the gene expression data. In this paper, we present a new clustering algorithm from an information-theoretic point of view. First, we propose the minimum entropy (measured on *a posteriori* probabilities) criterion, which is the conditional entropy of clusters given the observations. Fano's inequality indicates that it could be a good criterion for clustering. We generalize the criterion by replacing Shannon's entropy with Havrda-Charvat's structural α-entropy. Interestingly, the minimum entropy criterion based on structural α-entropy is equal to the probability error of the nearest neighbor method when $\alpha = 2$. This is another evidence that the proposed criterion is good for clustering. With a nonparametric approach for estimating *a posteriori* probabilities, an efficient iterative algorithm is then established to minimize the entropy. The experimental results show that the clustering algorithm performs significantly better than k-means/medians, hierarchical clustering, SOM, and EM in terms of adjusted Rand index. Particularly, our algorithm performs very well even when the correct number of clusters is unknown. In addition, most clustering algorithms produce poor partitions in the presence of outliers while our method can correctly reveal the structure of data and effectively identify outliers simultaneously.

1 Introduction

When the cell undergoes a specific biological process, different subsets of its genes are expressed in different stages of the process. The particular genes expressed at a given stage (*i.e.* under certain conditions) and their relative abundance are crucial to the cell's proper function. Technologies for generating high-density arrays of cDNAs and oligonucleotides enable us to simultaneously observe the expression levels of many thousands of genes on the transcription levels during important biological processes. Such global view of thousands of functional genes changes the landscape of biological and biomedical research. Large amounts of gene expression data have been generated. Elucidating the patterns hidden in these gene expression data is a tremendous opportunity for functional genomics.

A preliminary and common methodology for analyzing gene expression data is the clustering technique. Clustering is the process of partitioning the input data into groups or *clusters* such that objects in the same cluster are more similar among themselves than to those in other clusters. Clustering has proved very useful for discovering important information from gene expression data. For example, clustering can help identify groups of genes that have similar expression patterns under various conditions or across different tissue samples [6, 13]. Such co-expressed genes are typically involved in related functions. Clustering is also often the first step to discover regulatory elements in transcriptional regulatory networks [1, 27]. Co-expressed

genes in the same cluster are probably involved in the same cellular process and strong expression pattern correlation between those genes indicates co-regulation.

The clustering problem can be formally stated as follows: given a dataset of $\mathcal{X} = \{x_i | i = 1, \ldots, n\}$ and an integer $m > 1$, map $\mathcal{X}$ onto $C = \{c_j | j = 1, \ldots, m\}$ so that every x_i is assigned to one cluster c_j. As seen later, this definition is not so general as to be consistent with all the types of clustering strategies, e.g. hierarchical clustering and EM algorithm. For our purpose, however, it is adequate enough. Besides analyzing gene expression data, clustering can also be applied to many other problems, including statistical data analysis, data mining, compression, vector quantization, etc. As a branch of statistics, cluster analysis has been studied extensively in the literature. A large number of clustering algorithms have been proposed, such as hierarchical clustering, k-means/medians [7,15], expectation maximization (EM) algorithm [4], self-organizing maps (SOMs) [18], etc. In hierarchical clustering, a nested set of clusters is created. Each level in the hierarchy has a separate set of clusters. At the lowest level, each object is in its own unique cluster. At the highest level, all objects belong to the same cluster. The hierarchical clustering methods, though simple, often encounter difficulties with regard to the selection of merge or split points, which may lead to low-quality clusters if not well chosen at some steps [10]. Moreover, these methods do not scale well due to their high time and space complexity. In k-means, each object must belong to exactly one cluster. The k-means algorithm minimizes the sum of squared Euclidean distances between the objects in a cluster and the mean (*i.e.* center) of the cluster. The k-means method is sensitive to noise and outlier data points since a small number of such data can substantially influence the mean value [10]. Another disadvantage of the k-means method is that the user has to specify the number of clusters, which is often not known in practice. A slight variation of k-means is k-medians that uses medians instead of means. Compared with hierarchical clustering and k-means, the expectation maximization (EM) algorithm [4], a model-based method, plays with likelihood instead of distance. Besides, the EM algorithm does not compute actual assignments of objects to clusters, but *a posteriori* probabilities. In other words, each objects is found to belong to each cluster with a certain probability. It is interesting to note that the k-means algorithm can be seen as a special case of the EM optimization on a (spherical) Gaussian mixture model. It has been found in most instances that the EM algorithm has the advantage of reliable global convergence, low cost per iteration, economy of storage, and ease of programming. The problem with EM algorithm is the speed of convergence, which can be very slow in practice. Self-organizing map (SOM) [18] is also an important and widely used clustering method. SOM is a special class of artificial neural networks based on *competitive learning*, which has strong theoretical connection to the actual brain processing. The principal goal of SOMs is to transform an input object into a two-dimensional discrete map, and to perform this transformation adaptively in a topologically ordered fashion. However, it is very difficult to mathematically analyze the properties of SOM in a general setting.

Although these clustering algorithms are often applied to analyze gene expression data, they face two challenges in practice. Usually, these algorithms (except hierarchical clustering) require that the user specify the number of clusters in advance. In many situations, however, it is difficult for biologists to know the exact number of clusters since we may not know how many functional categories there exist or if some genes may belong to an unknown functional category. Although hierarchical clustering does not need the number of clusters, the user still needs decide how many groups and where to cut the tree of clusters after the clustering. Another problem with these algorithms is that they only perform well on "clean" data without outliers [7,15]. For instance, the k-means method is sensitive to outliers since a small number of such data can substantially influence the mean value [10]. However, outliers often contain important hidden information and provide clues for unknown knowledge. For example, a gene with abnormal expression may be related to some disease.

In this paper, we introduce an information-theoretic approach for clustering gene expression data. It is well-known that entropy is a measure of information and the uncertainty of a random variable. So, it is natural to employ the minimum entropy as a criterion for clustering. Besides, the concept on which entropy measures are based is similar to that of probabilistic dependence. Thus, we use entropy measured on *a posteriori* probabilities as the criterion for clustering. In fact, it is the conditional entropy of clusters given the observations. Thus, Fano's inequality indicates that the minimum entropy may be a good clustering criterion [3]. We also generalize the criterion by replacing Shannon's entropy with Havrda-Charvat's structural α-entropy [11]. Interestingly, the minimum entropy criterion based on structural α-entropy is equal to the probability error of the nearest neighbor method [5] when $\alpha = 2$. This indicates again that the minimum entropy could be a good criterion for clustering because the probability of error in the nearest neighbor

method is less than twice the Bayes probability of error [2]. With the minimum entropy criterion, the problem of clustering consists of two sub-problems (i) estimating *a posteriori* probabilities and (ii) minimizing the entropy. Although some parametric method could be employed to estimate *a posteriori* probabilities, we follow the nonparametric approach in this paper. A merit of the nonparametric approach is that we do not need much prior information (*e.g.* distribution) of the data. Since gene expression data usually have a complex structure, a particular choice of distribution may lead to a very poor representation of the data. In order to minimize the entropy, we propose an efficient iterative algorithm, which usually converges in a few steps. The experimental results on synthetic data and real gene expression data show that the new clustering algorithm performs significantly better than k-means/medians, hierarchical clustering, SOM, and EM in terms of adjusted Rand index [12]. In particular, our method performs very well even when the correct number of clusters is unknown. Besides, most clustering algorithms produce poor partitions in the presence of outliers while our method can correctly reveal the structure of data and effectively identify outliers simultaneously.

The rest of the paper is organized as follows. Section 2 introduces the minimum entropy criterion for clustering. A brief review of entropy and structural α-entropy is also included in this section. In Section 3, we follow the nonparametric approach to estimate *a posteriori* probabilities and propose an iterative algorithm to minimize the entropy. Section 4 describes the experimental results on both synthetic data and real data. Section 5 concludes the paper with some directions of future research.

2 The Minimum Entropy Criterion

In thermodynamics, entropy has important physical implications as the amount of "disorder" of a system. In information theory, the quantity entropy plays a central role as measures of information, choice and uncertainty. Mathematically, Shannon's entropy of a random variable X with a probability mass function $p(x)$ is defined as [26]

$$H(X) = -\sum_x p(x) \log p(x) \tag{1}$$

which is a functional of the distribution of X. It does not depend on the actual values taken by the random variable X, but only on the probabilities. Entropy is the number of bits on the average required to describe a random variable. In fact, entropy is the minimum descriptive complexity of a random variable [19].

Since entropy measures the amount of "disorder" of a system, we hope that each cluster has a low entropy because data points in the same cluster should look similar. Thus, we would like to employ some form of entropy in the objective function of clustering. A straightforward minimum entropy clustering criterion could be defined as

$$J = \sum_{j=1}^{m} p_j H(X|c_j) \tag{2}$$

where $H(X|c_j)$ is the entropy of cluster c_j, and p_j is the probability of cluster c_j such that $\sum_{j=1}^{m} p_j = 1$. Suppose each cluster c_j follows the d-dimensional Gaussian distribution with covariance matrix Σ_j. So, $H(X|c_j) = \log(2\pi e)^{d/2} + \frac{1}{2}\log|\Sigma_j|$ and

$$J = \frac{1}{2}\sum_{j=1}^{m} p_j \log|\Sigma_j| \tag{3}$$

by discarding additive constant $\log(2\pi e)^{d/2}$. This falls into the conventional minimum variance clustering strategy and seems a reasonable clustering criterion. However, it is actually not adequate for clustering because it neglects the semantic aspects of data. Note that the data contain some hidden *meaning*. That is, they refer to or are correlated according to some system with certain physical or conceptual entities. In clustering/classification, the semantic information that we care about is the category of objects. The task of machine learning is just to infer these categorical information. However, entropy was developed without the consideration of the meaning of data because the concept of entropy was developed originally for communication where the meaning of messages is irrelevant to the transmission of messages [26]. Besides, it was assumed during the development of entropy that the information source is *ergodic* [26]. It implies that the information source has only one statistical structure. In contrast, we naturally assume in cluster

analysis that the data are drawn from a *mixed* source made up of a number of pure components that are each of homogeneous statistical structure.

Based on these observations, we think that a good minimum entropy clustering criterion has to reflect the relationship between data points and clusters. Such relationship information helps us to reveal the meaning of data, i.e. the category of data. Besides, it also helps us to identify the components, i.e. clusters, of mixed information source. Since the concept on which entropy measures are based is similar to that of probabilistic dependence, we think that the entropy measured on *a posteriori* probabilities could be such a suitable quantity:

$$H(C|x) = -\sum_{j=1}^{m} p(c_j|x) \log p(c_j|x) \tag{4}$$

where C is the random variable of category taking values in $\{c_1, \ldots, c_m\}$ with probabilities $p_1, \ldots, p_m$. In (4), we compute *a posteriori* probabilities $p(c_j|x)$ to determine how much information has been gained. $H(C|x)$ is maximized when all $p(c_j|x)$ are equal. In this case, the object x could come from any source (i.e. cluster) equally probably, and thus we do not know which cluster the object x should belong to. This is also intuitively the most uncertain situation. On the other hand, $H(C|x)$ is minimized to 0 if and only if all the $P(c_j|x)$ but one are zero, this one having the value unity. That is, we are certain of the cluster of x only if $H(C|x)$ vanish. Thus, $H(C|x)$ can assess the dependence between x and C well.

By integrating x on the whole data space, we obtain the minimum entropy clustering criterion:

$$H(C|X) = -\int \sum_{j=1}^{m} p(c_j|x) \log p(c_j|x) p(x) dx \tag{5}$$

The above quantity is actually the *conditional entropy* of the random variable C given the random variable X [26]. The conditional entropy $H(C|X)$ measures how uncertain we are of C on the average when we know X. The conditional entropy has many interesting properties. It is known that

$$H(C|X) \leq H(C) \tag{6}$$

with equality if and only if X and C are independent. The quantity $H(C) = -\sum_{j=1}^{m} p_j \log p_j$ is the "pure" entropy of clusters, where p_j is the *a priori* probabilities of each cluster. Intuitively, Equation (6) says that knowing the random variable X can reduce the uncertainty in C on the average unless X and C are independent. This indicates that the minimum $H(C|X)$ could be a good clustering criterion. Besides, we know that $I(C;X) = H(C) - H(C|X)$ is the mutual information [26]. Thus, we may also use the mutual information $I(C;X)$ to evaluate the quality of clusters. In this paper, we confine ourselves to using conditional entropy as clustering criterion. In fact, the proposed method can be easily extended to the case with mutual information.

Fano's inequality [3] provides us another strong evidence that minimum $H(C|X)$ could be a good clustering criterion. Suppose we know a random variable X and we wish to guess the value of the correlated category information C. Fano's inequality relates the probability of error in guessing the random variable C to its conditional entropy $H(C|X)$. Suppose we employ a function $\hat{C} = f(X)$ to estimate C. Define the probability of error

$$P_e = \Pr\{\hat{C} \neq C\} \tag{7}$$

Theorem 1 (Fano's Inequality)

$$H(P_e) + P_e \log(m-1) \geq H(C|X) \tag{8}$$

This inequality can be weakened to

$$1 + P_e \log m \geq H(C|X) \tag{9}$$

Note that $P_e = 0$ implies that $H(C|X) = 0$. In fact, $H(C|X) = 0$ if and only if C is a function of X [3]. Thus, we can estimate C from x with zero probability of error if and only if C is a function of X. Fano's inequality indicates that we can estimate C with a low probability of error only if the conditional entropy $H(C|X)$ is small. In machine learning, P_e is expected to be small by the implicit assumption that X contains adequate information about C, i.e. $H(C|X)$ is small. Thus, the minimum $H(C|X)$ is a natural criterion for clustering.

2.1 Havrda-Charvat's Structural α-Entropy

So far, we have only considered Shannon's entropy. However, many measures of entropies have been introduced in the literature to generalize Shannon's entropy, *e.g.* Renyi's entropy [24], Kapur's entropy [16], and Havrda-Charvat's structural α-entropy [11], etc. We are particularly interested in the Havrda-Charvat's structural α-entropy for reasons that will be clear later. The structural α-entropy is defined as

$$H^{\alpha}(X) = (2^{1-\alpha} - 1)^{-1} \left[\sum_{x} p^{\alpha}(x) - 1 \right] \tag{10}$$

where $\alpha > 0$ and $\alpha \neq 1$. With different degrees α, one can obtain different entropy measures. For example, when $\alpha \to 1$, we obtain Shannon's entropy:

$$\lim_{\alpha \to 1} H^{\alpha}(X) = -\sum_{x} p(x) \log p(x)$$

When $\alpha = 2$, we have the quadratic entropy:

$$H^{2}(X) = 1 - \sum_{x} p^{2}(x) \tag{11}$$

In the above equations, we discard the constant coefficients for simplicity. With structural α-entropy, we get the clustering criterion:

$$H^{\alpha}(C|X) = 1 - \int \sum_{j=1}^{m} p^{\alpha}(c_j|x) p(x) dx \qquad \alpha > 1 \tag{12}$$

For $0 < \alpha < 1$, we can get the criterion by flipping the sign of the above formula. If the quadratic entropy is employed, we have

$$H^{2}(C|X) = 1 - \int \sum_{j=1}^{m} p^{2}(c_j|x) p(x) dx \tag{13}$$

Recall that, as a classification method, the nearest neighbor method has the following probability of error [5]:

$$R_{NN} = 1 - \int \sum_{j=1}^{m} p^{2}(c_j|x) p(x) dx \tag{14}$$

which is identical to Equation (13). Since the probability of error R_{NN} is less than twice Bayes probability of error [2], the minimum $H^{2}(C|X)$ could potentially be a good criterion for clustering because Bayes probability of error is the minimum probability of error over any other decision rule, based on the infinite sample set [9].

Another merit of structural α-entropy is that it satisfies the strong recursivity property. Suppose a random variable C has the distribution $P = (p_1, p_2, \ldots, p_m)$. In what follows, we write the entropy $H(x)$ as $H_m(p_1, p_2, \ldots, p_m)$. A measure of entropy $H_m(p_1, p_2, \ldots, p_m)$ will be said to have the recursivity property with respect to recursive function $g(p_1, p_2)$ if

$$H_m(p_1, p_2, \ldots, p_m) = H_{m-1}(p_1 + p_2, \ldots, p_m) + g(p_1, p_2) H_2 \left(\frac{p_1}{p_1 + p_2}, \frac{p_2}{p_1 + p_2} \right)$$

holds for all $m \geq 3$ [17]. Here $g(p_1, p_2)$ is a known continuous function. A stronger type of recursivity enables us to express H_m in term of two entropies, one of type H_{m-k+1} and the other of type H_k. In cluster analysis, the recursivity property is appreciated, especially when the data exhibit a nesting relationship between clusters. It is known that the only measures which satisfy the sum function property and the strong recursivity property are the Havrda-Charvat and Shannon's measures of entropy [17].

In summary, given a dataset $\mathcal{X} = \{x_1, \ldots, x_n\}$, we have the following minimum entropy clustering criterion

$$J = \begin{cases} 1 - \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} p^{\alpha}(c_j|x_i) & \alpha > 1 \\ -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} p(c_j|x_i) \ln p(c_j|x_i) & \alpha = 1 \\ \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} p^{\alpha}(c_j|x_i) - 1 & 0 < \alpha < 1 \end{cases} \tag{15}$$

3 The Clustering Algorithm

By employing (15) as the criterion, the problem of clustering consists of two sub-problems (i) estimating $p(c_j|x)$ and (ii) minimizing the entropy. In fact, these two problems cannot be clearly separated as we shall see later.

3.1 Nonparametric Estimation of *a Posteriori* Probability

To estimate $p(c_j|x)$, we could employ some parametric method. However, it may not be appropriate for clustering complicated gene expression data because the choice of any particular distribution could lead to a very poor representation of the data if the data has a complex structure. We therefore seek a nonparametric method for modeling the data. There are two kinds of nonparametric estimation techniques, *Parzen density estimation* [22, 25] and *k-nearest neighbor density estimate* [20]. They are fundamentally very similar, but exhibit some different statistical properties. In what follows, we give a brief overview of these two nonparametric density estimation methods.

Consider estimating the value of a density function $p(x)$ at a point x. We may set up a small window $R(x)$ (*e.g.* hyper-cube or hyper-ellipsoid) around x. Then, the probability mass of $R(x)$ may be approximated by $p(x) \cdot v$ where v is the volume of $R(x)$. On the other hand, the probability mass of $R(x)$ may also be estimated by drawing a large number (say n) of samples from $p(x)$, counting the number (say k) of samples falling in $R(x)$, and computing k/n. Equating these two probabilities, we obtain an estimate of the density function as

$$p(x) = \frac{k}{n \cdot v} \tag{16}$$

If we fix the volume v and let k be a function of x, we obtain Parzen density estimate. On the other hand, we may fix k and let v be a function of x. More precisely, we extend the region $R(x)$ around x until the kth nearest neighbor is found. This approach is called the k-nearest neighbor density estimate.

By Bayes's rule, we have

$$p(c_j|x) = \frac{p(c_j)p(x|c_j)}{p(x)}$$

We may use n_j/n as the estimator of $p(c_j)$, where n_j is the number of points in cluster c_j. If Parzen density estimate is employed, we have

$$p(c_j|x) = \frac{\frac{n_j}{n} \cdot \frac{k(x|c_j)}{n_j \cdot v}}{\frac{k(x)}{n \cdot v}} = \frac{k(x|c_j)}{k(x)} \tag{17}$$

Thus, the estimate of $p(c_j|x)$ is just the ratio between the number of samples from cluster c_j and the number

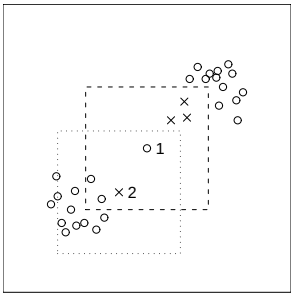


Figure 1: An illustration that the total entropy could increase when an object is assigned to the cluster containing most of its neighbors.

of all samples in the local region $R(x)$. The minimum entropy criterion turns out to be

$$J = \begin{cases} 1 - \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(\frac{k(x_i|c_j)}{k(x_i)}\right)^{\alpha} & \alpha > 1 \\ -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{k(x_i|c_j)}{k(x_i)}\log\frac{k(x_i|c_j)}{k(x_i)} & \alpha = 1 \\ \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(\frac{k(x_i|c_j)}{k(x_i)}\right)^{\alpha} - 1 & 0 < \alpha < 1 \end{cases} \tag{18}$$

If k-nearest neighbor estimate is used, we obtain

$$p(c_j|x) = \frac{\frac{n_j}{n}\cdot\frac{k}{n_j\cdot v(x|c_j)}}{\frac{k}{n\cdot v(x)}} = \frac{v(x)}{v(x|c_j)} \tag{19}$$

Similarly, we can get a corresponding clustering criterion.

3.2 An Iterative Algorithm

In this section, we try to develop a clustering algorithm to optimize the criterion (18). However, the criterion (18) (and the corresponding one with the k-nearest neighbor estimation) is not suitable for directly clustering the data because we can minimize $H(C|X)$ to 0 by simply putting all data points into one cluster. Such an optimal solution is trivial and interferes with finding the practically useful partitions. Thus, instead of directly clustering the data, we will present an iterative algorithm to reduce the entropy of an initial partition given by any other clustering methods (e.g. k-means). That is, the algorithm searches a local optimal solution starting from a partition given by some other method. This algorithm can be easily modified for optimizing the criterion with the k-nearest neighbor estimation. In principle, a (hill-climbing type) iterative algorithm starts with the system in some initial configuration. A standard rearrangement operation is applied to the system such that the objective function is improved. The rearrangement configuration then becomes the new configuration of the system, and the process is continued until no further improvement can be found.

In our case, an intuitive idea to update the partition is to assign a data object x to the cluster containing most of its neighbors. This reassignment actually decreases the entropy associated with x because any change toward unequalization of the probabilities decreases the entropy [26]. Suppose a point x is assigned to the cluster c_i currently and most neighbors of x are assigned to the cluster $c_j \neq c_i$. Moreover, suppose n_i neighbors of x belong to c_i and n_j neighbors of x belong to c_j such that $n_i < n_j$. After x is reassigned to cluster c_j, the difference between $n_i - 1$ and $n_j + 1$ is larger than that of n_i and n_j. So, we make the difference between the probabilities $p(c_i|x)$ and $p(c_j|x)$ larger, and thus reduce the entropy associated with x. Such an update, however, does not necessarily decrease the total entropy of the partition. Note that the

entropy associated with the neighbors of x also changes after the reassignment. Figure 1 gives an illustration that the total entropy of the partition could increase after the reassignment. In Figure 1, the dashed line box represents the window $R(x)$ around data object 1, which has four neighbors belonging to cluster c_j denoted by 'x' and three neighbors (including object 1 itself) belonging to cluster c_i denoted by 'o'. The window around data object 2 is represented by the dotted line box. Note that all neighbors of object 2 belong to cluster c_i. If we reassign object 1 to cluster c_j, the entropy associated with object 2 will increase because it has a neighbor in cluster c_j after the reassignment and thus the "disorder" of the neighbors increases. Similarly, the entropy associated with the three other objects in cluster c_j increases. So, the total entropy of partition could increase although the entropy associated with object 1 decreases. Based on this observation, we propose an algorithm that considers the change of entropy associated with all neighbors of x.

Algorithm 1 Minimum Entropy Clustering Algorithm

Input: A dataset containing n objects, the number of clusters m, and an initial partition given by some other clustering method.

Output: A set of at most m clusters that locally minimizes the conditional entropy.

Method:

```
1: repeat
2:   for every object x in the dataset do
3:     if the cluster c_j containing most of the neighbors of x is different from the current cluster c_i of x
       then
4:       h ← Σ_y (H'(C|y) − H(C|y))
         {where y are neighbors of x, and x is also regarded as the neighbor of itself. H(C|y) and H'(C|y)
         are the entropy associated with y before and after assigning x to the cluster c_j, respectively.}
5:       if h < 0 then
6:         assign x to the cluster c_j
7:       end if
8:     end if
9:   end for
10: until no change
```

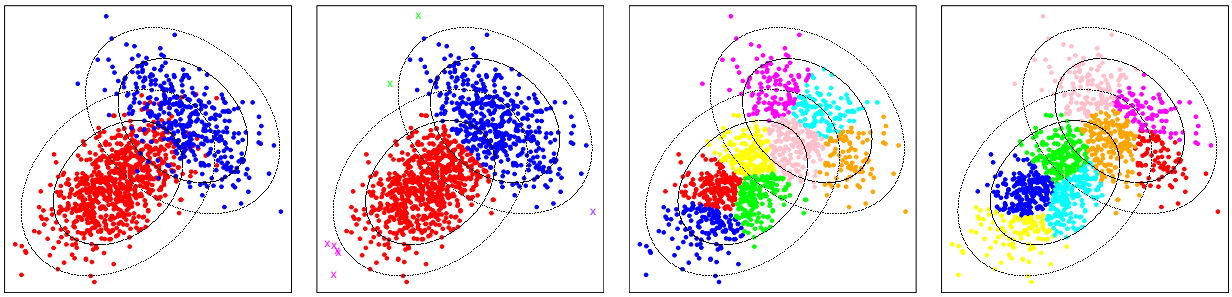
Theorem 2 *Algorithm 1 converges after a sufficient number of iterations.*

Proof. Clearly, the total entropy of the partition decreases in every step. Thus, the algorithm 1 converges since the entropy is bounded by 0. ■

Note that this algorithm could give a set of fewer than m clusters. The reason is that a cluster might migrate into another cluster to reduce the entropy during the iterations. This is different from most other clustering algorithms, which always return a given number of clusters. The speed of algorithm 1 depends on the number of iterations and the number of points that could be reassigned in each iteration, i.e. the points whose most neighbors belong to a different cluster. Usually, these kind of points exist in the overlap of clusters, and are only a small proportion of data. Besides, the number of neighbors could be regarded as constant in comparison with the whole data. So, the average time complexity of each iteration is usually less than $O(n)$ in practice. In the experiments, we found that the number of iterations is often very small, usually less than 20 for both synthetic and real data.

4 Experiments

We have tested our minimum entropy clustering (MEC) algorithm on both synthetic data and real gene expression data, in comparison with k-means/medians, hierarchical clustering, SOM, and EM, which are widely used for clustering gene expression data. For k-means/medians, hierarchical clustering, and SOM, we used Eisen's implementations [6]. For the EM algorithm, we used Fraley and Raftery's implementation [8]. To assess the quality of our algorithm, we need some objective external criteria. The external criteria could be the true class information, gene functional categories, *etc.* In order to compare clustering results against



(a) True class (b) MEC (c) k-means (d) EM

Figure 2: The synthetic two-component Gaussian data. Subfigure (a) represents the true class information. The left bottom component consists of 800 points. The upper right component consists of 400 points. The inner and outer contours are 2σ contour and 3σ contour, respectively. Subfigures (b), (c), and (d) are the clustering results when the specified number of clusters is 8. Different colors denote differnt clusters.

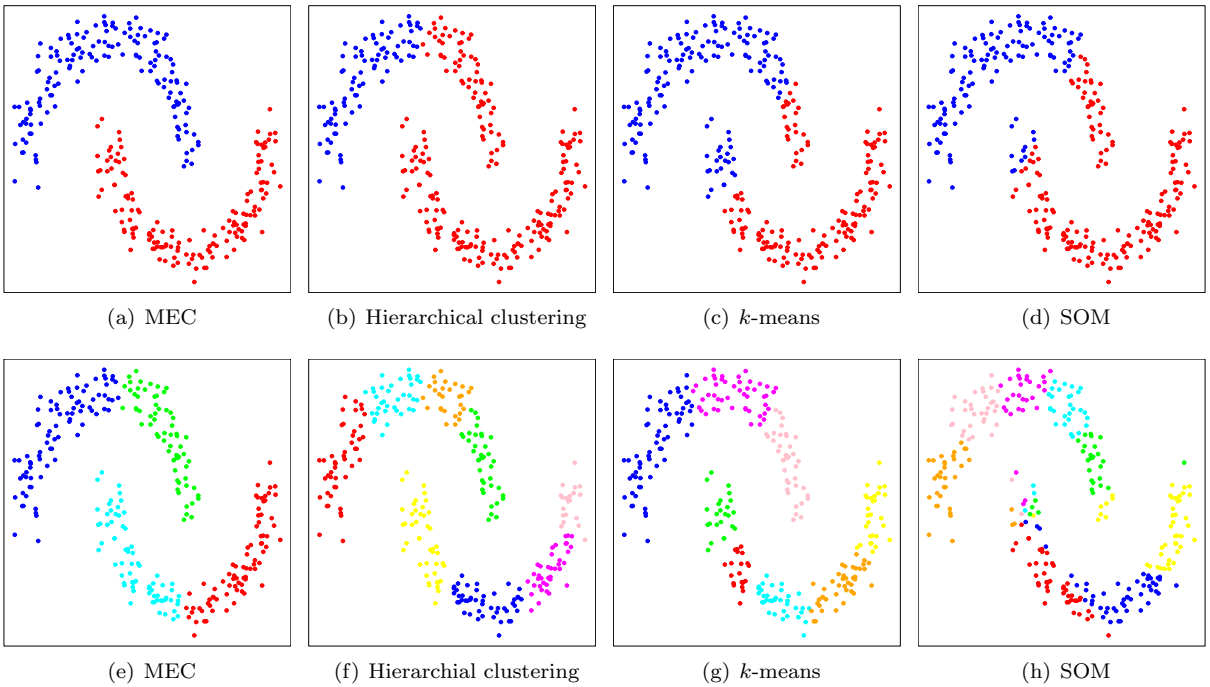
Table 1: Adjusted Rand index on the synthetic Gaussian data. The first column is the specified number of clusters. In k-means, the Euclidean distance is employed. The EM algorithm uses the Gaussian mixture model. The MEC algorithm uses the structural α-entropy with $\alpha = 2$.

m	k-means	EM	MEC
2	0.535	0.802	0.704
3	0.452	0.620	0.610
4	0.349	0.386	0.384
5	0.280	0.294	0.448
6	0.222	0.229	0.542
7	0.190	0.195	0.633
8	0.163	0.168	0.593
9	0.147	0.163	0.526
10	0.134	0.153	0.502

an objective external criterion, we employ adjusted Rand index [12] as the measure of agreement. Rand index [23] is defined as the number of pairs of objects that are either in the same group or in different groups in both partitions divided by the total number of pairs of objects. The Rand index lies between 0 and 1. When two partitions agree perfectly, the Rand index achieves the maximum value 1. A problem with Rand index is that the expected value of the Rand index between two random partitions is not a constant. This problem is corrected by the adjusted Rand index that assumes the generalized hypergeometric distribution as the model of randomness. The adjusted Rand index has the maximum value 1, and its expected value is 0 in the case of random clusters. A larger adjusted Rand index means a higher agreement between two partitions. The adjusted Rand index is recommended for measuring agreement even when the partitions compared have different numbers of clusters [21]. In this paper, the reported adjusted Rand index is averaged on 100 repeated experiments to reduce the influence of random initial partitions.

4.1 Synthetic Data

To give a visual illustration of our new method, we generate a two-dimensional synthetic data that consists of two components following the Gaussian distribution. The means and covariance matrices of the two components are [0.0, 0.0] and [2.0, 2.0], $\begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix}$ and $\begin{bmatrix} 1 & -0.3 \\ -0.3 & 1 \end{bmatrix}$, respectively. In Figure 2(a), we can see that these two components highly overlap. In this experiment, we compare our MEC algorithm and



(a) MEC (b) Hierarchical clustering (c) k-means (d) SOM

(e) MEC (f) Hierarchial clustering (g) k-means (h) SOM

Figure 3: The synthetic nonconvex data. The dataset contains two clusters generated by sine and cosine functions with a Gaussian noise of mean 0 and standard deviation 0.1. Each cluster consists of 150 points. The first and second rows are the clustering results when the specified number of clusters are 2 and 8, respectively. For k-means, hierarchical clustering, and SOM, Euclidean distance is used to measure (dis)similarity. For hierarchical clustering, we employ the complete link algorithm. For MEC, we use Shannon entropy.

the EM algorithm, both of which use the output of k-means as the initial partition.

Table 1 lists the adjusted Rand index achieved by k-means, EM and MEC algorithms. Both EM and MEC significantly improve the initial partitions given by k-means. It is not surprising that the EM algorithm obtains the best results when the specified number of clusters is correct since its model perfectly matches the data. However, we often do not know the exact number of clusters in practice. When the specified number of clusters are not correct, both k-means and EM algorithm have very low adjusted Rand indexes. In contrast, the MEC algorithm still performs very well. Figure 2 gives a graphical representation of the clustering results when 8 clusters are specified. Other configurations give similar results. Both k-means and EM return 8 clusters as specified and neither represent the structure of the data correctly. The MEC algorithm, however, returns 5 clusters and gives the correct structure of the data. More precisely, two of the five clusters contain most of the data objects and represent the structure of the data well. The remaining three clusters consist of 5, 2, and 1 objects (denoted by '$\times$'), respectively. These data objects stand apart from the bulk of the data. More precisely, they lie outside of the 3σ contour (*i.e.* the outer contour). Thus, these inconsistent data objects could be regarded as outliers. Many clustering algorithms are sensitive to the influence of outliers and can only perform well on "clean" data [7, 15]. However, outliers often contain important hidden information and provide clues for unknown knowledge. For example, a gene with abnormal expression may be related to some disease. Figure 2(b) shows that the MEC algorithm can effectively identify outliers and correctly discover the structure of the main data simultaneously.

In the above synthetic Gaussian dataset, the clusters have convex shapes. However, many clusters do not possess a convex shape in practice. In what follows, we test our method in comparison with hierarchial clustering method, k-means, and SOM on a synthetic nonconvex dataset. For k-means, hierarchical clustering, and SOM, Euclidean distance is used to measure (dis)similarity. For hierarchical clustering, we employ the complete link algorithm. For MEC, we use Shannon entropy. The dataset contains two clusters generated by sine and cosine function with a white noise of standard deviation 0.1. Each cluster consists of

Table 2: Adjusted Rand index on the yeast galactose data. The dataset contains 205 genes, which belong to four functional categories. For k-means/medians, hierarchical clustering, and SOM, both Euclidean distance and Pearson correlation coefficient are used to measure (dis)similarity. For hierarchical clustering, we employ the complete link algorithm. Both MEC algorithm and EM algorithm use the output of k-means with Euclidean distance as the initial partition. In the EM algorithm, we assume that clusters have the diagonal covariance matrices, but with varying volume and shape. For more general ellipsoidal setting, the EM algorithm meets the computational problem (singular covariance matrix).

		Specified number of clusters						
Method	Setting	4	5	6	7	8	9	10
MEC	Shannon's entropy	0.915	0.914	0.918	0.923	0.915	0.918	0.885
MEC	$\alpha = 2$	0.918	0.926	0.928	0.932	0.930	0.933	0.888
MEC	$\alpha = 3$	0.919	0.926	0.929	0.928	0.932	0.932	0.888
EM		0.788	0.716	0.661	0.629	0.593	0.557	0.547
k-means	Euclidean dist.	0.806	0.746	0.671	0.628	0.547	0.534	0.494
k-means	Pearson corr.	0.806	0.739	0.667	0.604	0.538	0.511	0.454
k-median	Euclidean dist.	0.823	0.748	0.648	0.618	0.552	0.520	0.470
k-median	Pearson corr.	0.756	0.672	0.608	0.576	0.513	0.461	0.410
SOM	Euclidean dist.	0.845	0.853	0.674	0.556	0.443	0.382	0.342
SOM	Pearson corr.	0.825	0.845	0.675	0.559	0.446	0.382	0.348
Hierarchical	Euclidean dist.	0.677	0.605	0.703	0.700	0.708	0.711	0.694
Hierarchical	Pearson corr.	0.677	0.605	0.703	0.700	0.708	0.711	0.694

150 points. The data is shown in Figure 3(a), which is also the clustering result of MEC when the specified number of clusters is 2. That is, MEC can perfectly cluster the dataset in this case. In contrast, all other three methods make some errors. When the specified number of clusters are not correct, our method can still return meaningful results. The second row of Figure 3 is the results when the specified number of clusters is 8. Similar results were obtained with other specified number of clusters. As shown in Figure 3(e), MEC returns four clusters in this case. Although it does not match the true clusters, it is still meaningful in practice. Note that the two reported clusters in each true cluster have very different patterns: one is up, the other is down. Thus, it is natural for us to interpret the dataset as four clusters. Since our method shows a clear superiority to other methods, we do not report the adjusted Rand index here to save space.

Table 3: The correlation coefficients between YLR316C, STO1 and the means of four clusters in the yeast galactose dataset.

Gene	Cluster	Correlation Coefficient	P-value
YLR316C	1	0.0538	0.8218
	2	0.1310	0.5819
	3	0.0608	0.7990
	4	0.2067	0.3818
STO1	1	0.0241	0.9198
	2	-0.0682	0.7750
	3	0.1117	0.6390
	4	-0.0913	0.7018

4.2 Real Gene Expression Data

We used two gene expression data to test the MEC algorithm. The first data is the yeast galactose data on 20 experiments [14]. For each cDNA array experiment, four replicate hybridizations were performed. Yeung *et al.* extracted a subset of 205 genes that are reproducibly measured, whose expression patterns reflect four functional categories in the Gene Ontology (GO) listings [28]. The dataset contains approximately 8% of missing data. Yeung *et al.* applied k-nearest neighbor ($k = 12$) to impute all the missing values. We used this subset of data in our test with the four functional categories as the external knowledge. [1] For each cDNA array experiment, we used the average expression levels of four measurements for cluster analysis. Before clustering, we normalized the data for each gene to have mean 0 and variance 1 across experiments.

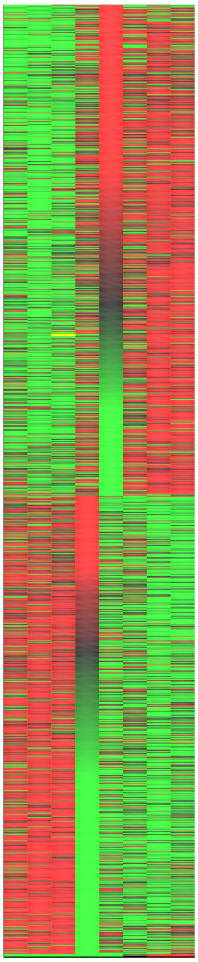
The experimental results are listed in Tables 2. Clearly, the MEC algorithm performs much better than k-means/medians, hierarchical clustering, SOM, and EM, especially when the specified number of clusters are far from the true number of clusters. Note that the EM algorithm even gives worse partitions than k-means in some cases. One possible reason is that such a small size data is not sufficient for EM to estimate its parameters, which are a lot more than those of k-means. Another reason is that the data may follow some unknown distribution so that the assumption of Gaussian mixture is not suitable. For yeast galactose data, the MEC algorithm achieves a very high adjusted Rand index (> 0.9). This indicates that the MEC algorithm is capable of effectively grouping genes in the same functional category according to their expression levels. Moreover, the MEC algorithm achieves a higher adjusted Rand index even when the specified number of clusters is larger than the correct number (*i.e.* 4 in this case). The reason is that, when the specified number of clusters is larger than the correct one, the MEC algorithm could use the "extra" clusters to identify outliers and thus improve the quality of the final partition. In this yeast galactose data, the MEC algorithm identified two genes as outliers. They are YLR316C and STO1, belonging to functional category 3 (nucleobase, nucleoside, nucleotide and nucleic acid metabolism). In the original dataset, STO1 is incorrectly classified as functional category 2 (energy pathways, carbohydrate metabolism, and catabolism). To verify that these two genes are really outliers in this microarray experiment, we calculate the Pearson's correlation coefficients between them and the means of clusters. The correlation and coefficients and corresponding P-values are listed in Table 3. Since Pearson's correlation coefficient follows a t-distribution of $n-2$ degrees of freedom (n is the number of arrays, e.g. $n = 20$ in this case), the statistical significance of a given correlation coefficient r is tested using a t-test with the hypotheses

$$H_0 : r = 0$$
$$H_1 : r \neq 0$$

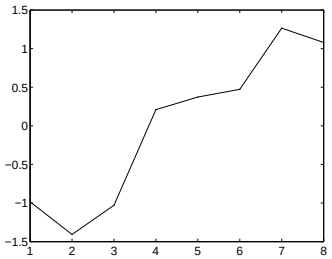
A low P-value for this test (say, less than 0.05) indicates that there is evidence to reject the null hypothesis. That is, there is a statistically significant relationship between the two variables. Since all correlation coefficients have very large P-values in our tests, it indicates that these two gene have very different expression patterns from the others. Thus, they can be regarded as outliers in this experiment. Note that the outliers are relative to the dataset. In a larger dataset, these two genes may not be outliers.

The above dataset is a typical gene expression dataset in which the expression levels of genes are measured at many time points or under different conditions to elucidate genetic networks or some important biological process. Another type of gene expression data evaluates each gene in a single environment but in different types of tissues, for example body atlas data or tumor data. In what follows, we will test our method on a prostate cancer gene expression data. Since the goal is to characterize gene expression in cancer cell lines rather than to elucidate some biological process, it is not suitable to use the functional categories of genes as external information. If we know if a gene is related to some specific tumor (or its subtypes), we may use this information to calculate adjusted Rand index. However, only few tumor-specific molecular markers are currently known by molecular oncology. Thus, we will not compute the adjusted Rand index on the tumor gene expression dataset when evaluating the method. In [29], Zhao *et al.* examined the gene expression profiles in androgen sensitive (AS) and androgen insensitive (AI) prostate cancer cell lines on a genome-wide scale. They measured the transcript levels of 27365 genes in five AS (LNCaP, LAPC-4, MDA PCa 2a, MDA PCa 2b, and 22Rv1) and three AI (PC-3, PPC-1, and DU145) prostate cell lines using cDNA microarrays.

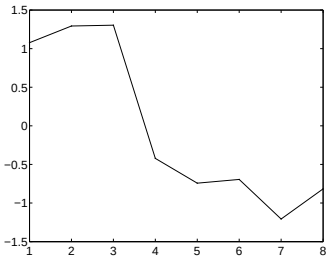
[1] We choose this subset rather than the whole dataset for experiments because it is very hard to determine the consistent functional categories for several thousand genes since may genes are multifunctional. Besides, it is also not easy to choose a suitable level of functional category for a large number of genes.



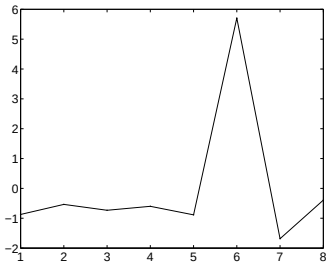
(a) Heatmap



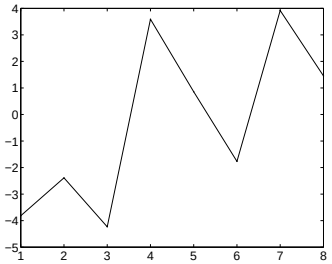
(b) AS



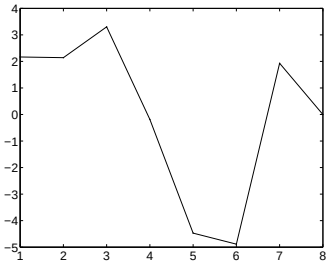
(c) AI



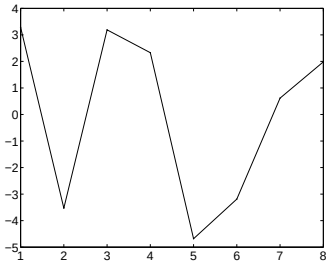
(d) IMAGE:2092597



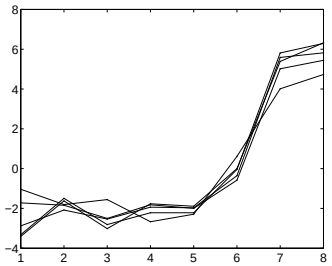
(e) IMAGE:2490680



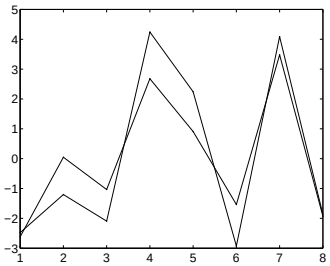
(f) IMAGE:669375



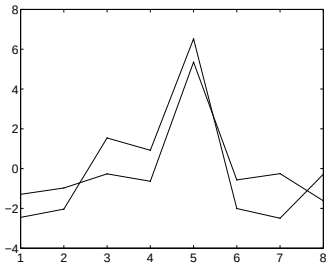
(g) IMAGE:2545705



(h) IMAGE:1554962, 1870935, 139009, 731442, and 840726



(i) IMAGE:81316 & 868838



(j) IMAGE:2391979 & 2489752

Figure 4: The clustering results on the prostate cancer cell lines dataset. In subfigure (a), each row represents a gene. The columns represent cell lines DU145, PPC-1, PC-3 (3 AI), 22Rv1, LAPC-4, LNCaP, MDA PCa 2a, and MDA PCa 2b (5 AS) from left to right. Subfigures (b) and (c) are the average expression levels of the top cluster (up-regulated across AS) and the bottom cluster (up-regulated across AI), respectively. Subfigures (d)–(j) are the expression levels of seven outlier groups, respectively. For subfigures (b)–(j), the horizontal axis is the cell lines in the same order as subfigure (a), and the vertical axis is the expression levels.

Table 4: The detected outliers in the prostate cancer cell lines dataset.

ID	Description
IMAGE:1554962	Homo sapiens transcribed sequence with weak similarity to protein ref:NP_060265.1 (H.sapiens) hypothetical protein FLJ20378 [Homo sapiens]; Hs.270149
IMAGE:1870935	FN1; fibronectin 1; Hs.418138
IMAGE:139009	FN1; fibronectin 1; Hs.418138
IMAGE:731442	COL4A3BP; collagen, type IV, alpha 3 (Goodpasture antigen) binding protein; Hs.21276
IMAGE:840726	Homo sapiens transcribed sequence with weak similarity to protein pir:S57447 (H.sapiens) S57447 HPBRII-7 protein - human; Hs.47026
IMAGE:81316	ARG99; ARG99 protein; Hs.401954
IMAGE:868838	HPGD; hydroxyprostaglandin dehydrogenase 15-(NAD); Hs.77348
IMAGE:2391979	DHRS2; dehydrogenase/reductase (SDR family) member 2; Hs.272499
IMAGE:2489752	PIP; prolactin-induced protein; Hs.99949
IMAGE:2092597	GAGE5; G antigen 5; Hs.278606
IMAGE:2490680	TRGV9; T cell receptor gamma variable 9; Hs.407442
IMAGE:669375	DKK1; dickkopf homolog 1 (Xenopus laevis); Hs.40499
IMAGE:2545705	CNN3; calponin 3, acidic; Hs.194662

In particular, they selected 1703 clones representing 1261 unique genes that varied by at least 3-fold from the mean abundance in at least two cell lines. We perform our method on the expression levels of these 1703 clones. Before clustering, we impute the missing values using the k-nearest neighbor method ($k = 10$). Figure 4(a) is the clustering results when the specified number of clusters is 20. Other configurations have the similar results. As shown in Figure 4(a), there are two big clusters in the dataset. For the top cluster (872 genes) in the figure, genes are overall up-regulated across AS cells but down-regulated across AI cells. On the other hand, 818 genes in the bottom cluster are overall down-regulated across AS cells but up-regulated across AI cells. However, we observe that genes in neither clusters follow a uniform pattern across cell lines 22Rv1 and LAPC-4. According to Zhao (private communication), both 22Rv1 and LAPC-4 were established from Xenograpfts. Human cancer cells were implanted into immune deficient mice, and tumors grown in mice were taken out and implanted into new mice again. After serial passages, the cells became immortal. Maybe events happened during this process make them different from other cell lines that were established not as xenograpfts. Besides these two big clusters, our method also found 7 small clusters containing 1, 2, or 5 genes, which could be regarded as outliers. These small clusters are plotted in Figures 4(d) – 4(j). As a comparison, we also plot the average expression levels of the two big clusters in Figure 4(b) and 4(c), respectively. Clearly, these outliers have either different patterns from those of the two big clusters or have very large (or very small) expression levels on some cell lines. A descriptions of these outliers is given in Table 4. Further analyses on the outliers is in progress.

5 Conclusion

From an information-theoretic point of view, we propose the minimum entropy criterion for clustering gene expression data. We also generalize the criterion by replacing Shannon's entropy with Havrda-Charvat's structural α-entropy. With a nonparametric approach for estimating *a posteriori* probabilities, an efficient iterative algorithm is established to minimize the entropy. The experimental results show that our new method performs significantly better than k-means/medians, hierarchical clustering, SOM, and EM, especially when the number of clusters is incorrectly specified. In addition, our method can effectively identify outliers, which is very useful in practice. It would be interesting to extend this new algorithm to a subspace clustering method for finding clusters with different subsets of attributes (*i.e.* the bi-clustering problem). Besides, the presented idea could also be applied to supervised learning, although this paper only focuses on unsupervised learning.

6 Acknowledgement

We thank Ka Yee Yeung and Hongjuan Zhao very much for providing the gene expression data and valuable discussion, and Yudong He and Xin Chen for reading the draft and providing valuable comments. This work was partially supported by NSF grants ACI-0085910 and CCR-0309902, and National Key Project for Basic Research (973) grant 2002CB512801.

References

[1] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide array. *Proceedings of the National Academy of Sciences of USA*, 96(12):6745–6750, 1999.

[2] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[3] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, 1991.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, 39(1):1–38, 1977.

[5] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.

[6] M. B. Eisen, P. T. Spellman, P. O. Browndagger, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of USA*, 95(25):14863–14868, 1998.

[7] B. S. Everitt, S. Landau, and M. Leese. *Cluster analysis*. Oxford University Press, New York, 4th edition, 2001.

[8] C. Fraley and A. E. Raftery. MCLUST: Software for model-based clustering, discriminant analysis and density estimation. Technical Report 415R, Department of Statistics, University of Washington, 2002.

[9] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 2nd edition, 1990.

[10] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco, 2000.

[11] J. Havrda and F. Charvat. Quantification method of classification processes: Concept of structural α-entropy. *Kybernetika*, 3:30–35, 1967.

[12] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[13] T. R. Hughes, M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Coffey, H. Dai, Y. D. He, M. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard, and S. H. Friend. Functional discovery via a compendium of expression profile. *Cell*, 102(1):109–126, 2000.

[14] T. Ideker, V. Thorsson, J. A. Ranish, R. Christmas, J. Buhler, J. K. Eng, R. Bumgarner, D. R. Goodlett, R. Aebersold, and L. Hood. Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, 292:929–934, 2001.

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

[16] J. N. Kapur. Generalised entropy of order α and type β. *The Mathematics Seminar*, 4:78–94, 1967.

[17] J. N. Kapur. *Measures of information and their applications*. John Wiley & Sons, New York, 1994.

[18] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, New York, 3rd edition, 2001.

[19] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer-Verlag, New York, 2nd edition, 1997.

[20] D. O. Loftsgaarden and C. P. Quesenberry. A nonparametric estimate of a multivariate density function. *Annals of Mathematical Statistics*, 36(3):1049–1051, 1965.

[21] G. W. Milligan and M. C. Cooper. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research*, 21:441–458, 1986.

[22] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[23] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of American Statistical Association*, 66:846–850, 1971.

[24] A. Renyi. On measures of entropy and information. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, volume 1, pages 547–561. University of California Press, 1961.

[25] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27(3):832–837, 1956.

[26] C. E. Shannon. A mathematical theory of communication. *Bell System Techical Journal*, 27:379–423 and 623–656, 1948.

[27] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, 1999.

[28] K. Y. Yeung, M. Medvedovic, and R. E. Bumgarner. Clustering gene-expression data with repeated measurements. *Genome Biology*, 4(5):R34, 2003.

[29] H. Zhao, Y. Kim, P. Wang, J. Lapointe, R. Tibshirani, J. R. Pollack, and J. D. Brooks. Genome-wide characterization of gene expression variations and dna copy number changes in prostate cancer cell lines. *The Prostate*, 2004.