Estimation Distribution of Algorithm for Fuzzy Clustering Gene Expression Data

Feng Liu[1], Juan Liu[1], Jing Feng[1], and Huaibei Zhou[2]

[1] Computer School of Wuhan University, Wuhan University
Wuhan, China
wolflf@126.com, liujuan@whu.edu.cn,
genefeng@mail.whu.edu.cn
[2] International School of Software, Wuhan University, Wuhan, China
bzhou@whu.edu.cn

Abstract. With the rapid development of genome projects, clustering of gene expression data is a crucial step in analyzing gene function and relationship of conditions. In this paper, we put forward an estimation of distribution algorithm for fuzzy clustering gene expression data, which combines estimation of distribution algorithms and fuzzy logic. Comparing with sGA, our method can avoid many parameters and can converge quickly. Tests on real data show that EDA converges ten times as fast as sGA does in clustering gene expression data. For clustering accuracy, EDA can get a more reasonable result than sGA does in the worst situations although both methods can get the best results in the best situations.

1 Introduction

With the rapid advancement of genome sequencing projects, microarrays and related high-throughput technologies have been key factors in the study of global aspects of biological systems. Generally speaking, gene expression data can be gotten from microarray experiments by readout of the mRNA levels of genes, which are conducted to address specific biological questions. The microarray experiments are usually carried on a genome with a number of different samples (conditions) such as different time points, different cells or different environmental conditions [1]. These data are always stored in a matrix, in which each row corresponds to a gene, each column corresponds to a condition, and each entry is a real number and denotes the expression level of the gene under the specific condition. The matrix can be denotes by X(G,C), where G is the set of genes and C is the set of samples. When gene expression data are analyzed, common pursued objectives are to group genes over all samples or cluster samples over all genes. The promise of these objectives is that the similar genes exhibit similar behaviors over all samples, or *vice versa*. This process is called clustering.

A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters [2]. In machine

L. Jiao et al. (Eds.): ICNC 2006, Part II, LNCS 4222, pp. 328 – 335, 2006.

learning, clustering is an example of unsupervised learning. Unlike classification, clustering and unsupervised learning do not relay on predefined classes and class-labeled training examples.

There exist a large number of traditional clustering algorithms based on methods of statistics such as partitioning methods, hierarchical methods, density-based methods, grid-based methods and model-based methods. At the same time, many natural computation and other intelligence algorithms, such as neural network, evolutionary computation, fuzzy logic, are introduced into cluster analysis. For an example, fuzzy c means (FCM) is a popular method of cluster analysis and is applied in many fields now.

Combining genetic algorithms and fuzzy logic, zhao *et al* presented a genetic algorithm for fuzzy clustering [3]. However, the behavior of evolutionary computation algorithms such as GAs depends on a large extent on associated parameters like operators and probabilities of crossover and mutation, size of the population, rate of generational reproduction, the number of generations, and so on. So Larranaga and Lozano[4] presented a new algorithm—Estimation of distribution algorithm (EDA)—to come over the disadvantages of genetic algorithm based on probabilistic graphical models. Based on these ideas, we adopt estimation of distribution algorithms for fuzzy clustering gene expression data, combining EDA and fuzzy logic. Experiments on real data show that our methods can outperform sGA both in convergence speed and accuracy of clustering samples.

The remainder of this paper is organized as follows: Model of fuzzy cluster is presented in section 2 and estimation of distribution algorithm is presented in section 3. We do some experiments on real data and discuss the results in section 4. In section 5, we make a conclusion.

2 Model of Fuzzy Cluster

Suppose $X=\{x_1,x_2,\ldots,x_n\}$ is the set of objects to be clustered and each object $x_i=\{x_{i1},x_{i2},\ldots,x_{is}\}$ in X has *s* attributes. The objective of clustering is to partition n objects in X into *c* groups $X_1,X_2,\ldots,X_c$, which satisfy: $X_1 \cup X_2 \cup \ldots \cup X_c=X$, $X_i \cap X_j=\Phi$ for each $1\le i\ne j\le c$.

Traditionally, each object only belongs to one group and each group contains at least one objects. This partition method is called hard partition or crisp partition. Set μ_{ik} to be the grade of membership that object k belongs to group i, then, for crisp partition, the membership function can be denoted as follows:

$$\mu_{ik}=\begin{cases}1 & x_k \in X_i \\ 0 & x_k \notin X_i\end{cases} \tag{1}$$

Ruspini E.[5] introduced the fuzzy logic into partition and define the membershipμ_{ik} to be a real number in [0,1] and satisfy that

$$E_f=\{\mu_{ik} \mid \mu_{ik}\in[0,1];\ \sum_{i=1}^{c}\mu_{ik}=1,\forall k;0<\sum_{k=1}^{n}\mu_{ik}<n,\forall i\} \tag{2}$$

Suppose $U=[\mu_{ik}]_{c\times n}$ is the partition matrix and $P=\{ p_i \mid p_i = (p_{i1},p_{i2},\ldots,p_{is})$ is clustering center (also called clustering prototype) of group i, i=1,2,…, c$\}$. According to Dunn [12] and Bezdek [13], the objective function of clustering is to minimize the following equation:

$$J(U,P)=\sum_{k=1}^{n}\sum_{i=1}^{c}(\mu_{ik})^{m}(d_{ik})^{2}\ \text{, s.t. } \mu_{ik}\in E_f \tag{3}$$

where d_{ik} is the distance between the k^{th} object and the i^{th} clustering center and m is weight coefficients. Here m is in $[1,\infty)$ and usually m=2.

According to Zimmermann, H.J[6], if clustering center p_i is known, then fuzzy partition matrix U can be computed as follows:

$$\mu_{ik}=\begin{cases}\dfrac{1}{\sum_{j=1}^{c}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m-1}}} & \text{if } I_k=\phi \\ 0,\ \forall i\in \overline{I_k} \text{ and } \sum_{i\in I_k}\mu_{ik}=1 & \text{if } I_k\neq\phi\end{cases} \tag{4}$$

Where $I_k=\{i|1\le i\le c, d_{ik}=0\}$ and $\overline{I}_k=\{1, 2, \ldots, c\}-I_k$.

In a word, fuzzy clustering is to find a fuzzy partition matrix U or a vector of clustering prototype (clustering center), which can minimize equation (3). In this paper, we try to find the best clustering prototype using estimation of distribution algorithms.

3 Estimation of Distribution Algorithm

3.1 Description of Algorithms

Genetic Algorithm is widely used in searching problems as an optimization algorithm. However, the researcher requires experiences in order to choose the suitable values for the parameters in the algorithm. Therefore, a new type of algorithms, Estimation of Distribution Algorithms (EDAs), was introduced [4]. The algorithms try to make easier to predict the movements of the populations in the search space as well as to avoid the need for so many parameters. Like GAs, the algorithms are also based on the populations and have a theoretical foundation on probability theory.

Unlike GAs, the new individuals in the next population are generated without crossover or mutation operators in EDAs. In fact, they are randomly reproduced by a probability distribution estimated from the selected individuals in the previous generation. At the same time, in EDAs the interrelations between the different variables representing the individuals are expressed clearly by means of the joint probability distribution associated with the selected individuals at each generation.

Suppose a population consists of *R* chromosomes (individuals) and each chromosome is composed of *n* genes as $X_1,X_2,\ldots,X_n$. Then a generic scheme of EDA approaches are shown as figure 1 and the essential steps are following:

Step 1. Generate randomly R individuals, composed of n-dimension, in D_0 generation.
Step 2. Select m ($m<R$) individuals, denoted by D_{h-1}^{m}, from the population in h-1 generation following a criterion.
Step 3. Induce the n-dimensional probabilistic model that better represents the interdependences between the n variables. The model can be presented by a directed acyclic graph (DAG). This is the most crucial step in EDA.
Step 4. Propagate R new individuals, which constitute the new population D_h, by carrying out the simulation of the probability distribution.

Steps 2, 3 and 4 are looped until the terminated condition is satisfied.

There are many ways to estimate the joint probability distribution associated with the selected individuals from the previous generation in discrete domains and more details are introduced in [4].

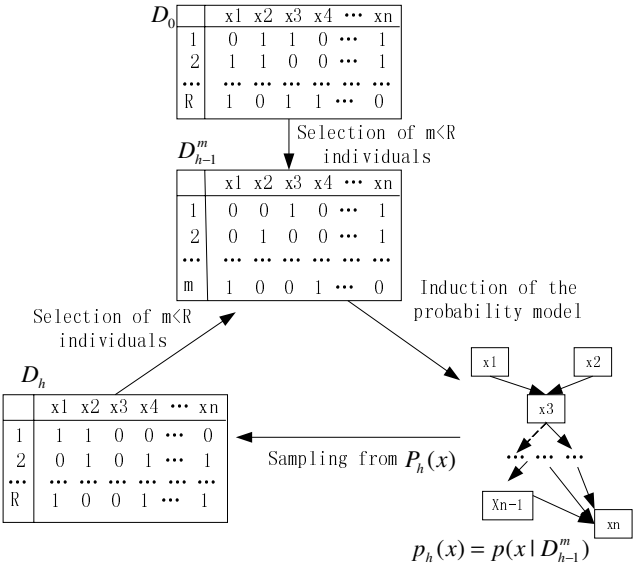


Fig. 1. Illustration of EDA approaches in the optimization process

3.2 Parameter Selection

Each individual in the population denotes clustering prototypes of each partition. Suppose that each object has s attributes and all objects are classified into c classes, then the chromosome is composed of $s*c$ real numbers, which ranges are between the minimum and maximum value of the attribute in all objects. Each real number is discretized into 20-length gray code. Thus each chromosome is coded into $s*c*$20-length gray code. The population size is set to 100 in this paper.

The crucial step in EDA is to induct the probability model from the last populations. In this paper, we suppose that the n-dimensional joint probability distribution factorizes like a product of n univariate and independent probability distribution. So

the famous algorithm UMDA (univariate marginal distribution algorithm) in [4] is adopted to form the directed acyclic graph (DAG) and propagate the next generation in our program.

We use the probabilistic logic sampling (PLS) proposed in [7] to sample the population in next generation. Stochastic Universal Sampling (SUS) is used to select the m individuals from the population in the previous generation. The generation gap in our method is set to 0.95 and the maximum generation is set to 100.

Equation (3) is computed for each individual and taken as the fitness function because our objective is to minimize the equation. In equation (3), d_{ik} is the Euclid distance between k^{th} object and i^{th} clustering center and the fuzzy partition matrix can be computed equation (4).

4 Experiments and Results

We tested our method using two gene expression data sets: ALL/AML data set and human renal tumor data set, which can be downloaded from Gene Expression Omnibus (GEO).

First, we filtered the genes with more than 20 percents null values over all samples. Then, just like [8], we only selected 50 genes that were most informative about the class distinction in the data for each data set. The genes were scored by the "twoing rules" using the tool package Rankgene, which is developed by [9].

The program is developed in MATLAB 7.01 for windows XP using GATBX toolbox, which is developed by University of Sheffield (1994).

To compare with sGA, we also did experiments on the same data sets using sGA. The probabilities of crossover operator and mutation operator were set to be default n GATBX toolbox and other parameters were set as same as UMDA's.

4.1 ALL/AML Data Set

The ALL/AML data set, which is published in [10], consists of the expression levels of roughly 6,800 human genes and data are measured using an Affymetrix oligonucleotide array from bone marrow samples collected from 47 patients suffering from acute lymphoblastic leukaemia (ALL) and 25 patients suffering from acute myeloid leukaemia (AML).

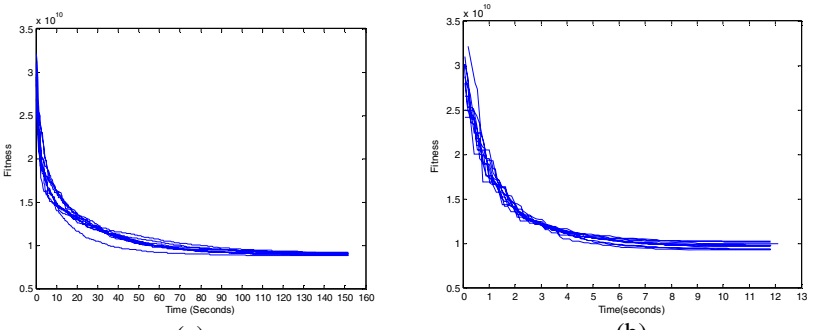
According to the types of patients, we grouped the samples into two clusters using different methods: EDA and sGA, and each run 10 times. The cluster details in the best situations and the worst situations are shown in table 1 and table 2, respectively.

Table 1. Best result for ALL/AML data

	Cluster No.	Cluster samples	AML	ALL	Cluster Error	Total Error
EDA	Cluster 1	24	24	0	0	1/72
	Cluster 2	48	1	47	1/48	
sGA	Cluster 1	24	24	0	0	1/72
	Cluster 2	48	1	47	1/48	

Table 2. Worst result for ALL/AML data

	Cluster No.	cluster samples	AML	ALL	Cluster Error	Total Error
EDA	Cluster 1	26	24	2	2/26	3/72
	Cluster 2	46	1	45	1/46	
sGA	Cluster 1	17	12	5	5/17	18/72
	Cluster 2	55	42	13	13/55	



(a) (b)

Fig. 2. Convergence speed for ten tests on ALL/AML data set. (a) GA's convergence curve (b) EDA's convergence curve.

From table 1, we can we can see that both methods reach the same error rate in the best situation, only one sample is grouped wrongly. The situation emerges 6 times in EDA and 4 times in sGA. In the worst situation listed in table 2 only 3 of 72 samples are wrongly grouped in EDA, while there are 18 of 72 wrongly grouped samples in sGA. That is to say, EDA are quite stable through all runs while the performance of sGA fluctuates across different runs. To sum up, EDA outperforms sGA in clustering ALL/AML gene expression data. Figure 2 shows the converge speeds of two methods. From figure 2, we can see EDA converges in around 5 seconds, while sGA converges in 60 seconds or so. Table 1,2 and figure 2 illustrate that, compared with sGA, EDA can converge more rapidly and get a more reasonable result in practice.

4.2 Kidney Cancer Data Set

Cutcliffe C. *et al* present a human renal tumor data set in 2005, which is composed of 22,283 genes and 35 samples [11]. Excluding three control samples, the 32 left samples come from two types of renal tumor patients: 18 of Wilms' tumor (WT) patients and 14 of clear cell sarcoma of the kidney (CCSK) patients. The expression data are measured using Affymetrix oligonucleotide arrays.

Performing the same experiments as in ALL/AML data set, we found that both methods can group the samples without errors in all runs, with result shown in table 3.

Table 3. Result for Kidney data in all situations

	Cluster No.	cluster samples	WT	CCSK	Error ratio	Total ratio
EDA	Cluster 1	14	0	14	0	0
	Cluster 2	18	18	0	0	
sGA	Cluster 1	14	0	14	0	0
	Cluster 2	18	18	0	0	

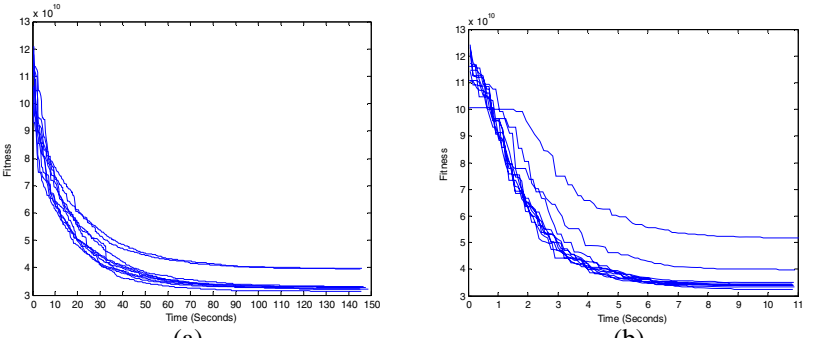


Fig. 3. Convergence speed for ten tests on kidney cancer data set. (a) GA's convergence curve (b) EDA's convergence curve.

The convergence speeds of both methods are shown in figure 3. From figure 3(a), we can find sGA converge slowly and usually converge in 80 seconds or so. However, EDA converge more quickly and can always converge in less than 8 seconds as Fig 3(b) shows. So we can conclude that, for kidney cancer data set, EDA outperforms sGA in convergence speed although both methods can get good result in clustering the samples using gene expression data.

5 Conclusions

In this paper, we put forwarded a fuzzy estimation of distribution algorithm to cluster gene expression data, which combines the estimation of distribution algorithm with fuzzy logic. Our method uses fewer parameters to reproduce the offspring than other evolutionary algorithms which can avoid adjusting the parameters in dealing with different problems. In order to evaluate our method, we clustered two real gene expression data sets using our methods and sGA. For ALL/AML data set, our method outperforms sGA both in accuracy and in convergence speed. For kidney cancer data set, both methods can get the perfect result, but EDA converges much more quickly than sGA. That is to say, in both real data, EDA outperforms sGA in convergence speed and gets a more reasonable result than sGA does. In this paper, we just assume that the variables are independent and use UMDA model to compute the joint

probabilities in EDA, in the future, we will consider the interrelations between variables and introduce more sophisticated joint probability models. Much more comparison experiments with other EAs are also needed to evaluate the overall performance of our method.

Acknowledgement

This paper is supported by the National Nature Science Foundation of China (60301009), Chenguang Project of Wuhan city (211121009).

References

1. Baldi, P., Hatfield, G.W.:DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modeling, Cambridge Univ. Press.(2002)
2. Han, J,. kamber, M.: Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers (2000):355-395.
3. Zhao, L. , Tsujimura, Y., Gen, M.: Genetic Algorithm For Fuzzy Clustering, International Conference on Evolutionary Computation (1996): 716-719
4. Larranga P., Lozano, J.A.: Estimation of Distribution Algorithms: a New Tool For Evolution Computation. Kluwer Academic Press, Boston, (2001)
5. Ruspini, E.: A New Approach to Clustering, Inf. Control, vol. 15 (1969):22--32
6. Zimmermann, H.J.: Fuzzy set Theory and Its Applications, 4thed. Kluwer Academic Publishers, (2001).
7. Henrion,. M: Propagation of Uncertainty In Bayesian Networks By Probabilistic Logic Sampling. In Uncertainty in Artificial Intelligence 2, Elsevier, North-Holland (1988): 149--163.
8. Murali, T. M., Kasif,S: Extracting Conserved Gene Expression Motifs From Gene Expression Data, In PSB, vol 8 (2003)
9. Su Y., Murali T.M., Pavlovic,V. *et al*: RankGene: Identification of Diagnostic Genes Based on Expression Data, Bioinformatics Vol. 19 No. 12 (2003): 1578-1579
10. Golub,T.R.: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science, Vol 286. No. 15 (1999):531-537
11. Cutcliffe, C., Kersey, D., *et al:* Clear Cell Sarcoma of The Kidney: Up-regulation of NeuralMarkers With Activation of The Sonic Hedgehog and Akt Pathways. Clin Cancer Res vol 11 No.22 (2005):7986-7994
12. Dunn,J.C., A graph theoretic analysis of pattern classification via Tamura's fuzzy relation. IEEE Trans. SMC, Vol 4,No.3 (1974): 310-313.
13. Bezdek,J.C., Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, (1981).