A Genetic Algorithm-Fuzzy-Based Voting Mechanism Combined with Hadoop Map-Reduce Technique for Microarray Data Classification

Ming-Tai Wu[1] Jain-Shing Wu[1] Chung-Nan Lee [1*] Ming-Cheng Chen[1]

[1] Department of Computer Science and Engineering, National Sun Yat-Sen University

Kaohsiung, Taiwan, ROC

{d953040015, d9134807}@student.nsysu.edu.tw, cnlee@mail.cse.nsysu.edu.tw,

m953040026@student.nsysu.edu.tw

Received 25 April 2013; Revised 15 May 2013; Accepted 16 June 2013

Abstract. Cloud computing is one of the major Information Technology (IT) trends that adopt IT maximum utility. It aids to analyze larger datasets for the hiding information. Existing methods may have a good performance, but it takes a lot of time to analyze microarray data. In this paper, we propose a novel Genetic algorithm (GA)-Fuzzy-based voting mechanism combined with the Hadoop to find the critical genes that affect the symptom. In addition, we proposed a voting mechanism adopted the Hadoop technique to increase the speed. Moreover, the proposed algorithm is also suitable for the Hadoop technique. Here, we used seven experimental datasets to verify the power of the proposed algorithm. The accuracies of four datasets using the proposed algorithm are better than the results obtained by the competing algorithm. However, there are three datasets are worse than the competing algorithm. Nevertheless, experimental results show that the proposed algorithm performs the best.

Keywords: genetic algorithm, fuzzy system, microarray, hadoop, cloud computing.

1 Introduction

Cloud computing [1] [5] is now one of the major Information Technology (IT) trends that adopt IT maximum utility. The main concept of cloud computing comes from the base of grid computing [2] [3], utility computing, cluster computing [4], and distributed systems in general. During the era when information is overflown, there are so many data that are hard to analyze. For example, the amount of message logs of famous web application, Facebook has about 500+ terabytes of data each day [6]. To analyze such large amount of data requires big storage and large amount of computation power. Hence, there are many researches that are developed for solving this problem. The clouding computing technique is developed for the analysis of the huge amount of data [7, 8].

The Map-Reduce technique [9] is one kind of cloud computing technique, and it first proposed by the Google Inc. The Hadoop Map-Reduce combines the Map-Reduce technique and Hadoop File System (HDFS). Map-Reduce concept is similar to the concept of "Divide and Conquer". In Hadoop Map-Reduce technique, the data is first uploaded to the Hadoop File System (HDFS). And then, the data are separated into many blocks. These blocks are mapped to a special key-value pair lists via the Map function on the Mapper. The Reducer collects all the key-value pairs generated from the Mappers and produces the final results.

In this paper, we use the program for finding the critical gene selection of microarray expression data as example to demonstrate the feature selection ability of the proposed method. Before describing the proposed method, some background information are given.

Cancer is a serious disease for human. To find the causes of cancer motivate more and more researchers to study in this topic. In other words, these causes are the hidden information and can be found by using data mining technique [10]. In addition, many technologies are developed to monitor and diagnosing the disease. The microarray technology is one of these technologies that can measure a large amount of the expression levels of genes at the same time. It is widely used in clinical oncology field. Up to now, the microarray expression profiles are treated as a large amount data, and become a data mining problem to find the useful information in this unprocessed dataset. Hence, researchers have proposed many algorithms to improve the classification of microarray data.

In general, the DNA microarray applications can be broadly classified into four primary categories: class discovery, class comparison, mechanistic studies, and class prediction [11]. In this paper, we focus on class prediction. The class prediction is to analyze the microarray expression data of unknown-class samples, and then classify these samples into known categories via the other well-known-class samples. If the prediction can be more precise, it can help the experts to diagnose the diseases.

*: corresponding author

In order to improve the accuracy of the class prediction, many algorithms have been proposed. The existing methods like hierarchical clustering, K-nearest-neighbor (KNN), linear discriminant analysis (LDA), support vector machines (SVM) are applied to classify microarray data. Although these classification methods can be used to classify microarray datasets for class prediction, they still have problems when dealing with the samples that are located at boundary between two classes. In this paper, we propose a GA-Fuzzy-Based voting mechanism to classify microarray datasets. The algorithm not only solves problems of boundary between classes, but also reserves advantage of existing methods. And then, the algorithm can find a membership function of each class in each gene from the given data. When all membership functions are found, using these results to vote for what class the sample of unknown classes belongs to.

The main contributions of the proposed algorithm are as the following. First, we integrate the GA-Fuzzy algorithm with Hadoop Map-Reduce technique to solve gene classification problem. It speeds up the GA-based fuzzy method to find better membership function. And then, we combine the upper bound α-Cut scheme with the Hadoop Map-Reduce technique to get better accuracy of each dataset through the voting mechanism. The proposed algorithm can blur the boundary between classes, and then reduce the error rate.

The remainder of this paper is organized as follows. In the following section, some articles for microarray analysis are shown. Definitions used in the proposed algorithm are described in Definition section. The proposed algorithm is described in Method Section. In the Experiment section, the experimental results are presented. After the Experiment section, we give some discussion in the following section. Finally, conclusions are drawn in the last section.

2 Related work

Up to now, there are many existing methods dealing with class prediction about microarray data. Pochet et al. [12] collected many methods and combined them to solve class prediction. The main methods are LS-SVM and FDA (is synonymous with LDA). And then, Pochet et al. combined LS-SVM with three different kernels, linear kernel, linear kernel (no regularization), and RBF kernel. FDA is combined with different PCA and different kernels, PCA (unsupervised PC selection), PCA (supervised PC selection), kernel PCA (kPCA) liner (unsupervised PC selection), kPCA liner (supervised PC selection), kPCA RBF (unsupervised PC selection), kPCA RBF (supervised PC selection). PCA was invented in 1901 by Karl Pearson [13]. It is used in exploratory data analysis and predictive models widely. The goal is to calculate the eigenvalue decomposition of a data matrix and eigenvector-based multivariate analyses for linear combination. By using this method, the dimensionality for linear combination can be reduced. Besides, ROC curve is also adopted to simulate selecting genes in [12].

Xiong and Chen [14] proposed a novel algorithm named "KerNN". "KerNN" is improved from KNN. The core of the algorithm is to use the gene selection method to determine whether the gene is used to be classified or not. The goal of the gene selection method is to select genes with the same class sample that is closer within-group and with the different class sample that is distant between groups. However, those methods have disadvantages such as lower accuracy in multi-class. In addition, the weighted voting [15] is one of a wide range of algorithms that have been used for microarray classification. The method [14] mentioned above will be compared with the proposed algorithm. In the next section, we give some essential definitions that are critical to the proposed GA-based fuzzy approach.

3 Essential Definition

Before describing the proposed algorithm, some essential functions are defined first. These functions are used to find out the maximum value of membership functions of different classes of the proposed GA-based fuzzy approach.

3.1 Density Function

Since sample numbers of each class in each dataset are not equal, weights for different classes are also different. In order to balance the weight of each class, the density function is proposed. In our work, the algorithm divides the solution space into several intervals first. Each interval contains the same number of sample for one class. For example, in our experiment, when algorithm produces the normal sample's membership function for a gene, the program divides 5 intervals in the solution space for normal samples and each interval contains the same number of normal samples. Suppose that a class A has $\left|P_{A_{total}}\right|$ elements, and $|P_A|$ is the set that the elements of class A within an interval, the density function $D(P_A)$ is

$$D(P_A) = \frac{|P_A|}{|P_{A_{total}}|}. \quad (1)$$

$|P_A|$: The number of elements of class A in an interval

$|P_{A_{total}}|$: The total number of elements in class A

An example of $D(P_A)$ is given in Figure 1. There is class *a* and class *b*. The weight of each point in class *a* is 1/8. There are two elements in the interval between two dash lines, the weight of elements of class *a* is 2/8 in the interval. Similarly, the total weight of elements of class *b* is 3/7 in the interval.

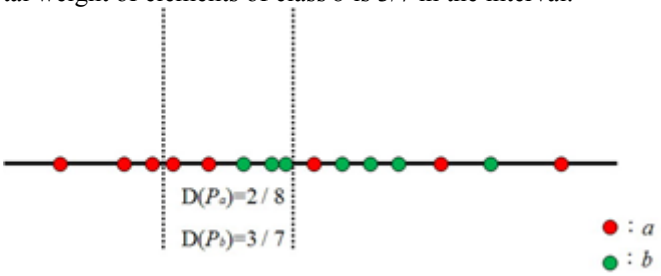


Fig 1. There is class *a* and class *b*. The weight of each point in class *a* is 1/8. There are two elements in the interval between two dash lines, the weight of elements of class *a* is 2/8 in the interval. Similarly, the total weight of elements of class *b* is 3/7 in the interval.

The value of density function is used to set the maximum number of each membership function that is described in the next section.

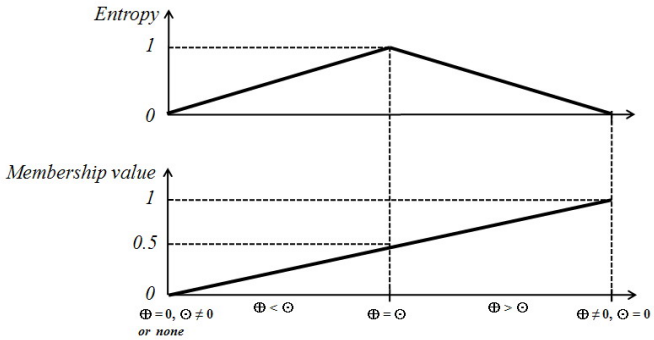
3.2 Transforming Function

In this paper, we adopted the entropy function to transform the value of density function to the maximum value of membership function. First, we used the value of density function instead of the real number of each sample to calculate the entropy value in each interval. The entropy function finds out the approximate shape of each membership function that belongs to each class. For interval *X*, if the entropy value ($H(X)$) is 1, the distributional probability of elements of two classes distributes evenly. At this time, the value of transforming function in the interval is 0.5. It means that the degree of the data for this class within the interval is 0.5. When the entropy value ($H(X)$) is close to 0 in the interval, the distributional probability of elements of certain class is larger than the other. The value of transforming function should approach to 1 when the data points in the interval *X* belong to the discussed class; otherwise the maximum value approaches to 0. The transforming function $T(X)$ is given below.

$$T(H(X)) = 0.5+0.5*(1- H(X))*(-1)^{1-k} \quad (2)$$

The transforming diagram from entropy to membership value is shown in Figure 2.

$$k = \begin{cases} 1 & \text{if the density value of discussed class is larger than the density value of the other classes} \\ 0 & \text{if the density value of discussed class is larger than the density value of the discussed classes} \end{cases} \quad (3)$$



⊕ : The density value of the discussed class

⊙ : The density value of the other classes

Fig 2. The transforming diagram from entropy to membership value

3.3 Coding of Membership Function

After obtaining the membership values of the intervals, we use them to generate the coding of membership function that are used in the GA-based fuzzy approach.

In the proposed algorithm, the shape of the membership function is trapezoid. The maximum value of these membership functions (individuals in the population) are set to the average value of transforming functions for each interval. The chromosome of the proposed GA-based fuzzy approach represents as:

C	α	β	γ	δ

where C is the center of the trapezoid (membership function), α is the distance between the left bottom (LB) point to left top (LT) point, β is the distance between C and the LT, γ is the distance between C and the right top (RT) point and δ is the distance between the right bottom (RB) point to RT. An example is shown in Figure 3.

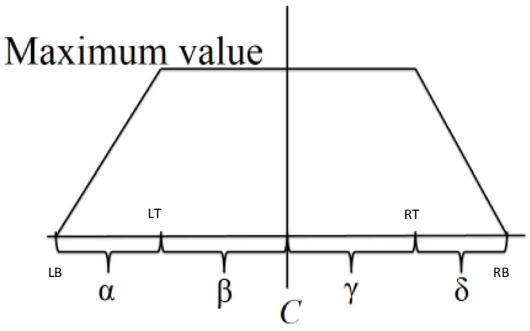


Fig 3. An example of membership function with chromosome (C, α, β, γ, δ)

4. Proposed Method

The proposed algorithm is divided into two phases: the learning phase and voting phase and described as the Figure 4.

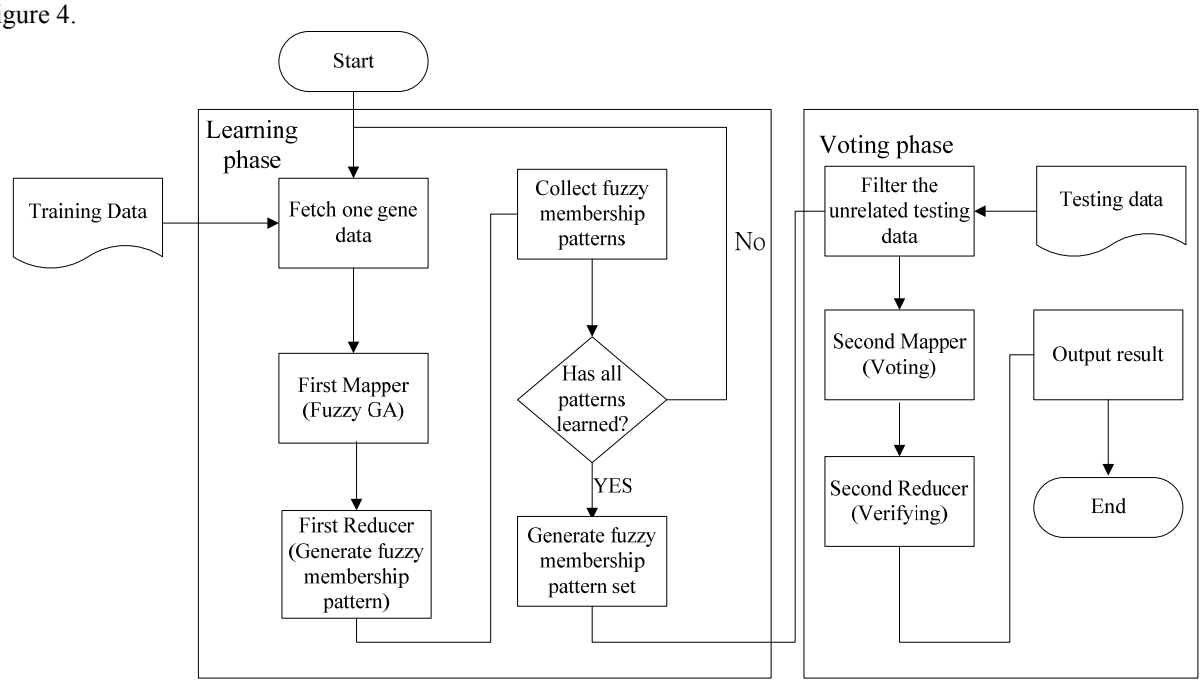


Fig 4. The flowchart of the proposed algorithm.

Before performing the proposed algorithm, the inputted microarray data should be fixed to eliminate the null value or error values. And then, the proposed algorithm transposes the microarray data to the format that can be used in the proposed algorithm.

4.1 Learning Phase

In the learning phase, the proposed fetches the gene data of all samples. The first Mapper receives the gene data and generates the individuals used in the GA-based fuzzy approach. The design of the fitness function is described as the following:

$$f = \frac{1}{\#N_{total}^{+}} * \sum_i S_i - \frac{1}{\#N_{total}^{-}} * \sum_j S_j \quad \textbf{(4)}$$

where

$$S_i = \begin{cases} M(i) & \text{if the } i\text{th data in the shape} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

and

$$M(i) = \begin{cases} MV & \text{if the ith data located within LT and RT} \\ \dfrac{\alpha - (|\,\text{Data}_i - \text{center}\,| - \beta)}{\alpha} \times MV & \text{if the ith data located within RT and RB} \\ \dfrac{\delta - (|\,\text{Data}_i - \text{center}\,| - \gamma)}{\delta} \times MV & \text{if the ith data located within LT and LB} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

MV is the maximum fuzzy value of the fuzzy function; $\#N_{total}^{+}$ is the number of data that belongs to the discussed class; $\#N_{total}^{-}$ is the number of data that belongs to the other class. The roulette method is used to select chromosomes to the next generation. Here, if the fitness value of one individual is less than or equal to zero, this individual is ignored in the selection.

In this paper, the max–min–arithmetical (MMA) crossover proposed in [15] is used to perform crossover to make the algorithm find an appropriate membership function and then converge quickly. In addition, the uniform mutation is adopted in the proposed GA-based fuzzy approach.

When the proposed algorithm gets the normal and abnormal membership functions of one gene produced by the GA-based fuzzy approach, the training data is used to calculate the corresponding fuzzy values. Ideally, if one datum belongs to normal, then its fuzzy value of normal membership function should be larger than the values of abnormal membership function; otherwise, the predicted class of this data is incorrect. Therefore, when the ratio of correct classification for training data to all training data at this gene is lower than a predefined threshold, then this gene is ill-judged. The proposed method ignores this gene in voting phase. The proposed algorithm generates the key value pairs for the membership functions of each gene with high accuracy as follows

<key, value> = <gene #, <c_1,α_1,β_1,γ_1,δ_1, c_2,α_2,β_2,γ_2,δ_2> >

where the c1,α1,β1,γ1, and δ1 is the normal chromosome; the c2,α2,β2,γ2,δ2 is the abnormal chromosome. The first Reducer receives all key-value pairs and returns them to the voting phase.

4.2 Voting Phase

In the voting phase, the proposed algorithm receives well-judged genes obtained from the learning phase. And then, according to these genes, the proposed algorithm filters out the unrelated genes of the testing data.

Every well-judged gene can give a normal fuzzy value and abnormal fuzzy value for a testing sample. Finally, the approach sums all normal fuzzy values and abnormal fuzzy values by using all well-judged genes. If the summation of all normal fuzzy values of one testing sample is greater than the abnormal one, this testing sample belongs to the normal class; otherwise it belongs to the abnormal class.

Since, the proposed algorithm using the membership functions belongs to different classes in the vote mechanism, the function can predict the class that the testing data belongs to. However, there are some noises that would mislead the predicted results. Hence, we use the technique α-cut to filter out the unrelated genes.

α-cut is one of basic concepts of fuzzy sets and used to "cut" fuzzy sets. The value of α is from 0 to 1. If an element of a fuzzy set mapping to fuzzy value is less than α, this element is ignored. On the contrary, if an element of a fuzzy set map to fuzzy value is larger than or equal to α, this element is accepted. Then, these accepted elements form a new fuzzy set. In this paper, the α-cut is used to filter the noise genes and find out the key genes to obtain the true voted results. If one data point P locates within the membership shape of one class C, then this point can claim that it belongs to that class.

However, it is possible that the membership function of each class owns different membership value. Figure 5 shows an example for the problem of α-cut.

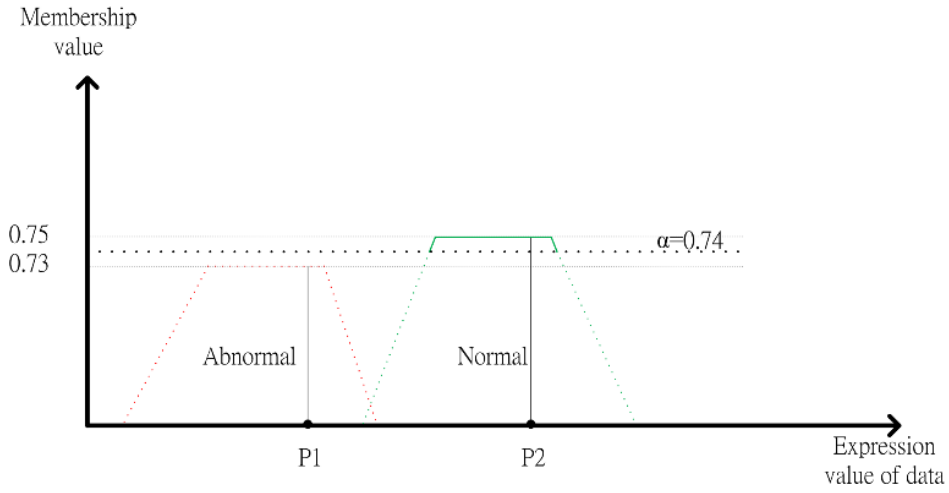


Fig 5. An example for the problem of α-cut

In Figure 5, when α is set to 0.74, and the fuzzy value of point $P1$ in the membership function which is in the abnormal class is 0.73. However, 0.73 is less than 0.74, point $P1$ cannot vote. Hence, the proposed would predict a wrong class which point $P1$ belongs to. On the other hand, the fuzzy value of membership function in the normal class of the point $P2$ is 0.75. Since, this fuzzy value is greater than 0.74, point $P2$ can vote, which means that point $P2$ belongs to normal. Hence, those points within abnormal range cannot be used to vote, but those points in normal range can vote. It makes the voting mechanism unfair and may be biased to the normal class, since the differences between these membership shapes are very large. In order to solve this problem, we proposed a method based on α-cut called "Upper bound α-cut".

Upper bound α-cut is to allow the samples within the main interval of the classes to vote. The core of the upper bound α-cut is to give up the overlap region which is covered by the entire other class. The region in others interval is omitted for the abnormal class, too. If the membership shape is lower than the α-cut, the vote value is the α-cut value; otherwise, the vote value is the fuzzy value of the correspond point P. We use Figure 6 and Figure 7 as examples.

Figure 6 shows ideal membership shapes of the abnormal class and the normal class. As Figure 6 shows, the abnormal data points are labeled in red color, and the data points belong to normal class are labeled in yellowish-brown color. However, these shapes can't be achieved. We can only obtain the raked trapezoid shape just like the shapes in Figure 7.

When the value of α-cut is greater than 0.65, the shape of the abnormal membership is omitted. However, using the upper bound α-cut, the three points of right side are within the main interval of abnormal membership shape. Hence, according to the upper bound α-cut, the vote values of these 3 points are set to the value of α-cut.

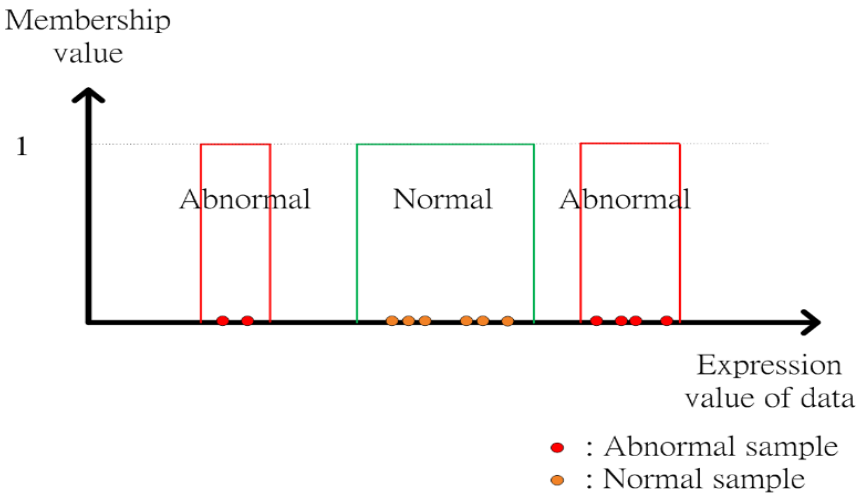


Fig 6. An example for ideal membership shapes

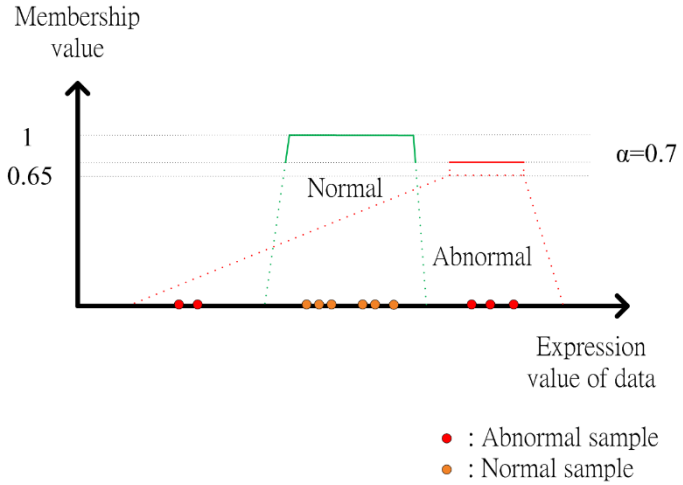


Fig 7. An example for the upper bound α-cut

After obtaining the voted results, each sample of the testing data is sent to the second Mapper to generate new key-value pairs:
<key, value>=<gene #, [1,0]> or
<key, value>=<gene #, [0,1]>
where the [1,0] means that this testing sample votes to the normal class and [0,1] means that this testing sample votes to the abnormal class. The second Reducer collects the voted results generated by the second Mappers, and then outputs the predicted classification results. If the predicted results are the same or similar to the ground truth of the testing samples, it means that the membership functions we found are critical to the diseases.

5. Experimental Results

The datasets used to verify the proposed algorithm are listed as Table 1. The experiments were implemented in Java on a cloud environment that the number of node is 4. Each node uses Intel Core 2 Quad 6600 CPU and 4 GB of RAM. The core of the resampling method is to expand the data three times for the training set. And then, use the resampling training set to learn by applying the proposed algorithm. The summary of the results of the numerical experiments on eleven cancer classification problems are given in Table 2, comparing five classification algorithms, the proposed algorithm (upper bound α-cut) in voting phase on training and test set.

Table 1. The information of datasets.

Dataset name	ALLAML	CNS	Colon	Lung	Lymphoma	Ovarian	Prostate
Sample #	72	60	62	181	77	253	102
Attribute #	7129	7129	2000	12533	7129	15154	12600

Table 2. Comparisons among the classification results using different gene selection methods for nine datasets, AMLALL, Colon, Lung, Ovarian, CNS, Lymphoma, and Prostate.

Experiments	KNN	ULDA	DLDA	SVM	KerNN	The proposed algorithm
AMLALL	3.32(1.21)	3.08(1.09)	2.95(0.78)	2.70(0.00)	2.70(0.00)	1.68(1.86)
CNS	19.52(5.88)	12.26(7.04)	22.42(5.58)	13.35(7.52)	15.32(5.60)	15.24(2.60)
Colon	14.03(3.76)	16.84(6.14)	12.65(4.58)	11.84(4.28)	11.58(4.97)	6.61(3.22)
Lung	1.21(0.98)	0.81(0.73)	0.47(0.57)	0.53(0.61)	0.31(0.54)	0.20(0.42)
Lymphoma	2.05(2.58)	2.05(2.09)	6.23(2.88)	1.03(1.59)	1.90(2.05)	5.21(1.58)
Ovarian	0.74(0.87)	0.02(0.13)	1.58(0.81)	0.17(0.42)	0.01(0.08)	0.70(0.38)
Prostate	7.41(2.47)	5.22(2.99)	6.73(3.02)	4.86(2.77)	4.90(2.53)	4.68(1.11)

Here, we follow the evaluated method in [6] to calculate Score A(Score B). A is the average error rate and B is the standard deviation. Then, the global performance of a classifier can be roughly evaluated in terms of the average score. Experimental results show that in these seven datasets, the proposed algorithm obtains four ad-

vanced results in those datasets. Although the other three datasets do not obtain the best results, the error rates are still within the acceptable range.

In Figure 8, we show the comparison of the computation time about the proposed methods verifying all datasets between the cloud environment and the single personal computer.

Figure 8 shows that the performance of the proposed algorithm in Hadoop environment can efficiently reduce a lot of computation time. It helps to speed up the redundant learning time for the proposed architecture.

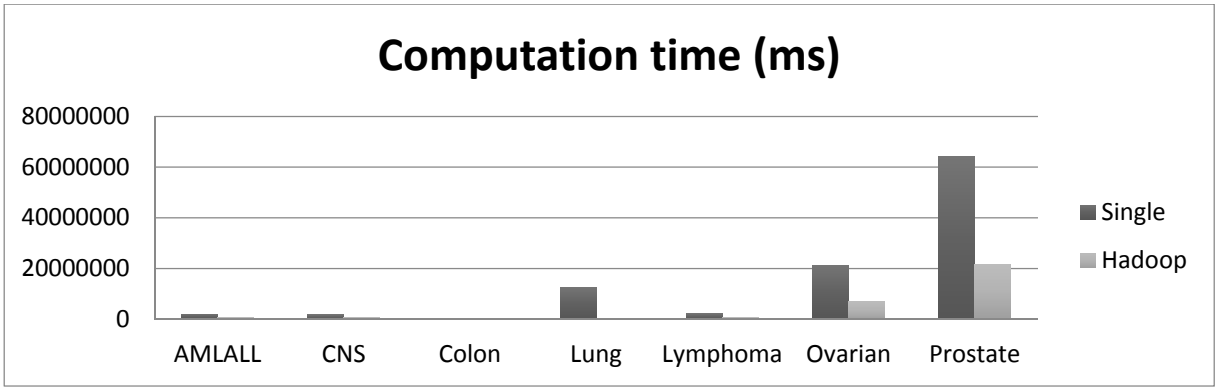


Fig 8. The computation time of a single computer and the Hadoop system in each datasets

6. Conclusions

We have proposed a GA-based fuzzy approach combined with Hadoop Map-Reduce technique and voting algorithm to classify two classes of microarray datasets. The experimental results indicate that when comparing with the competing methods in many kinds of datasets, the error rates can be reduced by using the proposed algorithm. Comparing with previous algorithms for microarray dataset classification problem, the proposed algorithm can easily be applied in the Hadoop environment, and it can decrease computation time. Based on experimental results, we can find that the proposed algorithm performs the best in many datasets.

In order to select genes, we proposed one voting mechanisms and the upper bound α-cut. Based on experimental results, using the upper bound α-cut obtains better accuracy.

In the future, we can apply this method to find more causative genes and using these genes to obtain hidden connections between diseases and drugs.

References

[1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Ratterson, A. Rabkin, I. Stoica, M. Zaharia, "A View of Cloud Computing," *Communications of the ACM*, Vol. 53, pp. 50-58, 2010.

[2] I. Foster, Y. Zhao, I. Raicu, S. Lu, "Cloud Computing and Grid Computing 360-degree Compared," in *Proceedings of 2008 Grid Computing Environments Workshop* (*GCE'08*), IEEE Press, pp. 1-10, 2008.

[3] S. Zhang, X. Chen, S. Zhang, X. Huo, "The Comparison between Cloud Computing and Grid Computing," in *Proceedings of 2010 International Conference on Computer Application and System Modeling* (*ICCASM*), IEEE Press, pp. V11-72-V11-75, 2010.

[4] H. Brunst, W.E. Nagel, A.D. Malony, "A Distributed Performance Analysis Architecture for Clusters," in *Proceedings of IEEE International Conference on Cluster Computing* (*CLUSTER'03*), IEEE Press, pp. 73-81, 2003.

[5] C. Gong, J. Liu, Q. Zhang, H. Chen, Z. Gong, "The Characteristics of Cloud Computing", in *Proceedings of 2010 IEEE International Conference Parallel Processing Workshops* (*ICPPW'10*), IEEE Press, pp. 275-279, 2010.

[6] J. Constine, "How Big Is Facebook's Data? 2.5 Billion Pieces Of Content and 500+ Terabytes Ingested Every Day," 2012, Available at:

http://techcrunch.com/2012/08/22/how-big-is-facebooks-data-2-5-billion-pieces-of-content-and-500-terabytes-ingested-every-day/

[7] H.F. Zhu, T.H. Liu, D. Zhu, H. Li, "Robust and Simple N-Party Entangled Authentication Cloud Storage Protocol Based on Secret Sharing Scheme," *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 4, No. 2, pp. 110-118, 2013.

[8] B.R. Chang, H.F. Tsai, C.M. Chen, "Evaluation of Virtual Machine Performance and Virtualized Consolidation Ratio in Cloud Computing System," *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 4, No. 3, pp. 192-200, 2013.

[9] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Communications of the ACM*, Vol. 51, pp. 107-113, 2008.

[10] C.W. Lin, T.P. Hong, C.C. Chang, S.L. Wang, "A Greedy-based Approach for Hiding Sensitive Itemsets by Transaction Insertion," *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 4, No. 4, pp. 201-227, 2013.

[11] J. Quackenbush, "Computational Approaches to Analysis of DNA Microarray Data," *Methods of Information in Medicine*, Vol. 45, pp. 91-103, 2006.

[12] N. Pochet, F.D. Smet, J.A.K. Suykens, B.L.R. De Moor, "Systematic Benchmarking Of Microarray Data Classification: Assessing the Role of Non-Linearity and Dimensionality Reduction," *Bioinformatics*, Vol. 17, pp. 3185-3195, 2004.

[13] K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space," *Philosophical Magazine*, Vol. 2, pp. 559-572, 1901.

[14] H.L. Xiong, X.W. Chen, "Kernel-based Distance Metric Learning for Microarray Data Classification," *BMC Bioinformatics*, Vol. 7, pp. 299, 2006.

[15] F. Herrera, M. Lozano, J.L. Verdegay, "Fuzzy Connectives Based Crossover Operators to Model Genetic Algorithms Population Diversity," *Fuzzy Sets and Systems*, Vol. 92, pp. 21-30, 1997.