35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

A High Performance Cloud Computing Platform for mRNA Analysis

Feng-Seng Lin, Chia-Ping Shen, *Student Member*, *IEEE*, Hsiao-Ya Sung, Yan-Yu Lam, Jeng-Wei Lin, and Feipei Lai, *Senior Member, IEEE*

Abstract— **Multiclass classification is an important technique to many complex bioinformatics problems. However, their performance is limited by the computation power. Based on the Apache Hadoop design framework, this study proposes a two layer architecture that exploits the inherent parallelism of GA-SVM classification to speed up the work. The performance evaluations on an mRNA benchmark cancer dataset have reduced 86.55% features and raised accuracy from 97.53% to 98.03%. With a user-friendly web interface, the system provides researchers an easy way to investigate the unrevealed secrets in the fast-growing repository of bioinformatics data.**

I. Introduction

The challenge in the field of biology is the enormous amount of existing data, which is complex and disordered. It is hard for people to sort and classify. With the growth of bioinformatics, it can use the computer science technology, methods and algorithms to analyze the huge biological data and explore the unrevealed secrets in these huge data. Since the biological data is huge and complex, it is hard to run and handle the overall data and features. Therefore, feature selection and classification method have become crucial technologies in this field.

Several successful attempts had been made to leverage the power of parallel computing to address the issue of huge computing demand in biomedicine data analyses. Systems in cloud computing architecture expectedly have a large amount of computing power, disk storage, and network bandwidth, and are scalable to handle dynamic computation demand. In [1], Ekanayake *et al*. had presented their experiences in applying two cloud technologies, Apache Hadoop [2] and Microsoft DryadLINQ [3-4], to bioinformatics applications, including a pairwise Alu sequence alignment application [5] and an Expressed Sequence Tag (EST) sequence assembly program [6]. To our best knowledge, there is not a system to handle both performance and cost. In this study, we proposed a two-layer-based Hadoop system of Genetic algorithms (GA) added Support Vector Machines (SVM) to speed up the training time, but not decrease the accuracy.

F. S. Lin and H. Y. Sung is with the Department of Computer Science and Information Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, 10617, Taiwan, R. O. C.

C. P. Shen, Y. Y. Lam, and F. P. Lai are with the Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, 10617, Taiwan, R. O. C.

J. W. Lin is with the Department of Information Management, Tunghai University, No.181, Sec. 3, Taichung Port Rd., Xitun Dist., Taichung City 40704, Taiwan, R. O. C.

*Correspondence to: Mr. F. S. Lin, phone: 886-958775761; e-mail: r98922096@csie.ntu.edu.tw

Support Vector Machines (Vapnik 1995), which are based on statistical learning theory and the structural risk minimum principle, are widely used due to the high accuracy and flexibility in modeling diverse sources of data [7]. It can minimize the upper bound of generalization error and seek the generalization model. However, the accuracy of multiclass classification degrades very rapidly as increase the number of classes. Some of the features will also interrupt the accuracy of SVM. Therefore, feature selection plays an important role in high dimensional biomedical data analysis.

Since feature selection is to select the most important and effective feature subset, it can reduce the time consumption and feature space dimension. It also can be seen as a process to find the optimal solution. We use Genetic algorithms (GA) as our feature selection tools, which is newly random search and optimization algorithms. GA provides an analogy of evolutionary process to solve problems in engineering and evaluate the individuals of each generation by their fitness. The few and selected individuals will continue to recombine and mutate to reproduce better offspring. Hence, GA is appropriate for solving the complex problem and nonlinear problems that the traditional search algorithms cannot solve well. The combination of GA and SVM can make for great process in multiclass classification performance. However, it will cause a challenge because the huge computation demands.

To deal with the huge computation demands, we use cloud computing architecture to parallel the computing process in Figure 1. Systems in cloud computing architecture expectedly have large amount of computing power, disk storage, and network bandwidth to handle the computation demands. In conclusion, we build up a novel Hadoop architecture to reduce the training time and improve the performance.

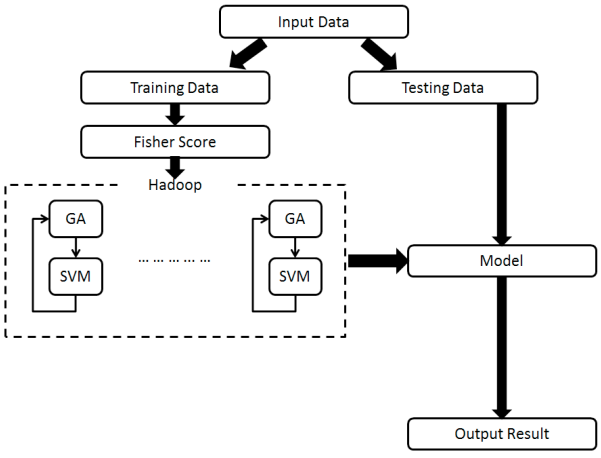

Fig. 1. System Architecture

II. METHOD

A. Fisher Score

In [8-9], it had been shown that the use of a low cost Fisher score on a sparse and high dimensional space can result in a compact and dense representation, which is desirable for image classification and retrieval problems.

As defined in (1), $\boldsymbol{I(p, q, k)}$ is used to evaluate the importance of a specific feature $\boldsymbol{k}$ ($1 \le \boldsymbol{k} \le \boldsymbol{n}$), for a hyper-plane $\boldsymbol{\Omega_{p,q}}$ corresponding to class $\boldsymbol{p}$ and class $\boldsymbol{g}$, where $\boldsymbol{\mu_{p,k}}$ and $\boldsymbol{\sigma_{p,k}}$ denote the mean value and standard deviation of the feature $\boldsymbol{k}$ for all training samples in class $\boldsymbol{p}$.

$$\boldsymbol{I(p, q, k)} = \frac{(\boldsymbol{\mu_{p,k}} - \boldsymbol{\mu_{q,k}})^2}{\boldsymbol{\sigma^2_{p,k}} + \boldsymbol{\sigma^2_{q,k}}} \quad (1)$$

$\boldsymbol{I(p, q, k)}$ aims at evaluating the differentiation capability between the two classes $\boldsymbol{p}$ and $\boldsymbol{g}$ as well as the stability in each class for a feature $\boldsymbol{k}$. For each hyper-plane, the importance of each feature is calculated first, and then certain features are selected according to their importance. The selected features are usually different for different decision hyper-planes.

B. Multiclass SVM Classification with GA

One approach for multiclass classification is to perform a one-against-one (OAO) classification. For two classes $\boldsymbol{p}$ and $\boldsymbol{q}$, a hyper-plane $\boldsymbol{\Omega_{p,q}}$ is determined by a standard SVM($\boldsymbol{p}$, $\boldsymbol{q}$) classification. When there are $\boldsymbol{m}$ classes in the dataset, this will result in $\boldsymbol{m}$*($\boldsymbol{m}$-1)/2 standard binary SVM classifiers. Each of these classifiers votes for a class for a testing or unknown sample $\boldsymbol{x}$. The number of votes for $\boldsymbol{x}$ in class $\boldsymbol{p}$ is calculated as (2).

$$\boldsymbol{v}(\boldsymbol{x}, \boldsymbol{p}) = |\{\boldsymbol{\Omega_{p,q}} \mid \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\Omega_{p,q}}) > 0\}| \quad (2)$$

The classification is processed by a max-wins strategy, as shown in (3).

$$\text{class}(\boldsymbol{x}) = \arg\max_{\boldsymbol{p}}(\{\boldsymbol{v}(\boldsymbol{x}, \boldsymbol{p})\}) \quad (3)$$

Applying feature selection before training a SVM classifier plays an important role in biomedical data analysis. To find a better $\boldsymbol{\Omega_{p,q}}$ for two classes $\boldsymbol{p}$ and $\boldsymbol{q}$, in this study, a GA (genetic algorithm) is applied to select the features and configuration to be used in SVM($\boldsymbol{p}$, $\boldsymbol{q}$) classification.

A GA mimics the process of natural evolution to produce useful solutions to optimization and search problems [10]. In a population, there are many individuals. Every individual represents a SVM model, and has its respective fitness, i.e., the accuracy of the SVM model. Since RBF kernel is used, in addition to the selected features, the accuracy also depends on the values of penalty ($\boldsymbol{C}$) and gamma (γ). For an individual $\boldsymbol{g}$, a standard binary SVM classification SVM($\boldsymbol{p}$, $\boldsymbol{q}$, $\boldsymbol{g}$) is invoked, and results in a SVM model with its fitness $\boldsymbol{u(p, q, g)}$. If the fitness is high, we think the individual has good genes, i.e., proper selections of $\boldsymbol{C}$, γ, and features, for SVM($\boldsymbol{p}$, $\boldsymbol{q}$) classification. The GA evaluates the fitness of all individuals in a generation, and picks good individuals to produce next generation by techniques inspired by natural evolution, e.g., mutation and crossover. When the GA terminates, the individual with the highest fitness, i.e., the most accurate SVM model, is chosen for SVM($\boldsymbol{p}$, $\boldsymbol{q}$). The GA can be summarized in four steps, as following.

1) Generate the initial population $\boldsymbol{M_1}$ of individuals randomly.
2) Compute the fitness $\boldsymbol{u(p, q, g)}$ for each individual $\boldsymbol{g}$ in the current population $\boldsymbol{M_k}$. $\boldsymbol{u(p, q, g)}$ is the accuracy of the SVM($\boldsymbol{p}$, $\boldsymbol{q}$, $\boldsymbol{g}$) classification when the corresponding features indicated by $\boldsymbol{g}$ are used.
3) Generate $\boldsymbol{M_{k+1}}$ by selecting good individuals from $\boldsymbol{M_k}$ to produce the offspring via genetic operators. The selection probability for an individual $\boldsymbol{g}$ in $\boldsymbol{M_k}$ is designed to be proportional to $\boldsymbol{u(p, q, g)}$.
4) Return to step 2 until a satisfying condition is reached.

An individual $\boldsymbol{g}$ uses three chromosomes to separately encode the penalty ($\boldsymbol{C}$), gamma (γ), and selected features that will be used in SVM($\boldsymbol{p}$, $\boldsymbol{q}$, $\boldsymbol{g}$). Standard binary representations are adopted for $\boldsymbol{C}$, and γ. As well, $\boldsymbol{g[i]}$ is 1 means the inclusion of the $\boldsymbol{i}$-th feature, whereas 0 indicates the exclusion of this feature.

When the GA terminates, the most accurate SVM model is chosen, as shown in (4).

$$\boldsymbol{\Omega_{p,q}} = \boldsymbol{\Omega_{p,q,g^*}}, \text{ where } \boldsymbol{g^*} = \arg\max_{\boldsymbol{g}}(\{\boldsymbol{u(p, q, g)}\}) \quad (4)$$

If the GA evolves $\boldsymbol{R}$ generations, the number of standard SVM invocations is $|\boldsymbol{M_1}|*|\boldsymbol{M_2}|*\ldots*|\boldsymbol{M_R}|$, where $|\boldsymbol{M_k}|$ is the number of individuals in $\boldsymbol{M_k}$.

C. Apache Hadoop Framework

Hadoop is open source software under the Apache Software Foundation [11] that it depends on Google File System and Google MapReduce. This framework is designed as a distributed file system and parallel computing architecture called HDFS (Hadoop Distributed File System).

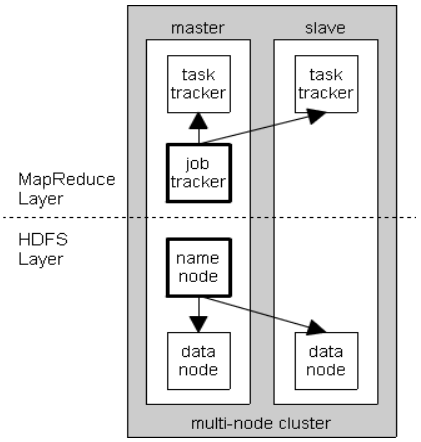

Fig. 2. Client Server Model of Hadoop

In statistics, a large cluster of thousands nodes and hard disk may have hard disk damaged or power supply breakdown. Thus, HDFS is designed for reliability to handle the hardware failure. And it is designed for big data and capable for handle Terabytes level data. In addition, A HDFS is composed of one

name node server and many data node servers in Figure 2. Name node server hosts the file system namespace, which operates data access, including moving, copying, deleting, opening or closing file. Data node servers host the storing of physical file blocks by HDFS block protocol across servers.

D. *Two Layer Cloud Computing*

As mentioned in above section, client server model of Hadoop divided into two layers. One is MapReduce layer, and another one is HDFS layer. In [10], we proposed paralleling job list by different hyper-planes. In real case, the training time in each hyper-plane depends on GA-SVM numbers. Thus, in this study, we proposed a two-layer architecture in Figure 3. The first layer is hyper-planes layer (HPL), and second layer is GA-SVM distributed layer (GDL). HPL is segmented by different hyper-planes, and each hyper-plane can divided into second layer by different GA-SVM task. Using this architecture, system can save the training time of idle hyper-plane due to early converge.

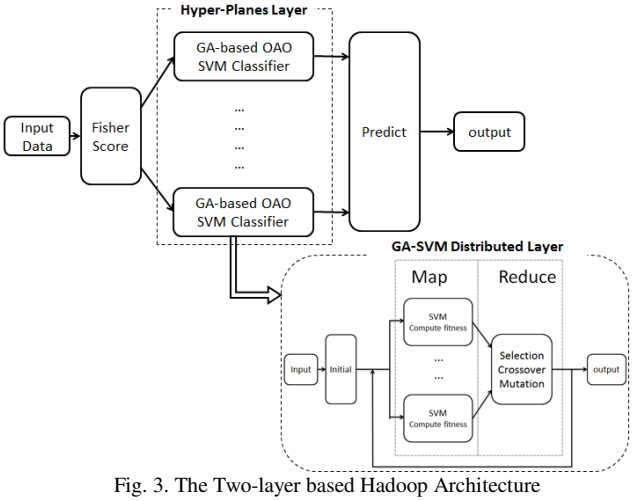

Fig. 3. The Two-layer based Hadoop Architecture

III. RESULTS AND DISCUSSION

In this paper, we selected an mRNA benchmark dataset as our data to evaluate the performance of removing one part of features. This dataset of mRNA gene is contained 144 training set and 54 test samples, which separated into 14 differential diagnosis of cancer. Each data contains 16,063 feature genes and expresses sequence tag for each sample. There are two type results experienced by these two data sets in the following statement.

A. *Time evaluation*

In Figure 5, we use the mRNA data sample to evaluate the performance of different amount of SVM under the same GA parameters settings. For each hyper-plane, the GA generated 100 individuals in the first generation and reserved the best 100 individuals for each generation. It would be terminated evolution when the algorithm repeats 200 times. According to the Figure 5, it is clear to see that the each map of original GA architecture has 100 tasks to finish; it should take more than 9 hours to finish the tasks. However, the new architecture could take fewer tasks than the original architecture. Due to it can parallel process the task and reduce the tasks that each maps should finish, it can improve the performance of genetic algorithm and reduce the cost time, as shown in Figure 5. In addition, the number of GA-SVM means that how many GA-SVM task distributed in second layer.

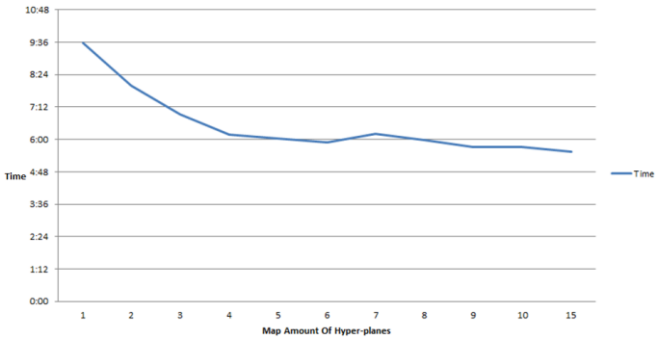

Fig. 4. Genetic Algorithm Execution Time

B. *Remove Feature*

Although genetic algorithm is strong randomness and stable, it takes too much time to converge. Moreover, features don't always have the beneficial effect on the accuracy. Some of the features are useless or even harmful. Therefore, we implement fisher score and some reduce methods to enhance the GA algorithm. We use fisher score to sort the features and separate these into ten parts. We use these ten parts to become some of the initial chromosomes, which could have better starting point of the algorithm. If the chromosomes don't have significant improvement during a period of time, the best individual of the generations will separate the features into certain parts. By erasing the separated part, it can generate the new chromosomes into the generation. If the accuracy of new chromosome doesn't decrease, we will pick this chromosome as a selected chromosome.

In experiments, we use mRNA data, which has 91 hyper-planes and 16,063 features, to experiment the removing feature method. The GA would generate 300 initial individuals for each hyper-plane and reserve the top 35 chromosomes for each generation. If the top 35 have no significant improvement during the 15 successive generations, the removing method will be triggered. It will separate the best chromosome into twenty parts. It will produce new twenty individuals by erasing one part of the separated parts. In Figure 5, we can see that the average features amount by using the original method is 2,097. By using the removing method, we reduce the feature amount into 282 (reduce 86.55% features). It also improves the total accuracy because it can get rid of the features which will lower the accuracy. Thus, we improved the accuracy from 97.53% into 98.03%. It was a significant improvement on the previous results in the literature [12-13] (76% ~ 90.96%). As shown in Table I, 51 test samples in 11 classes were perfectly classified, and only 1 test samples were falsely classified. The experiment result shows the effectiveness of our approach.

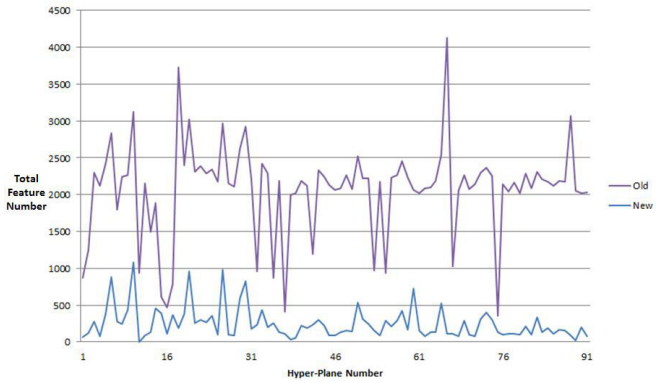

Fig. 5. Hyper-planes Feature Amount

IV. CONCLUSION

In this paper, we present a two-layer based cloud computing framework that exploits possible parallelism to speed up training analyses. In this framework, we had built a multiclass SVM classification tool with feature selection by GA. The evaluation of this tool on an mRNA benchmark cancer dataset showed that the accuracy of classification increased to 98.03% and computing time significatly reduced from 9.58 hours to 4.53 hours when 15 backend servers were used. This demonstrates the effectiveness and efficiency of the proposed cloud computing framework.

ACKNOWLEDGMENT

The authors would also like to thank Prof. Chih-Jen Lin and his research team members for providing the LIBSVM tool.

REFERENCES

[1] J. Ekanayake, T. Gunarathne, and J. Qiu, "Cloud Technologies for Bioinformatics Applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, pp. 998-1011, 2010.
[2] Apache Hadoop. Available: http://hadoop.apache.org/core/
[3] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, "Dryad: distributed data-parallel programs from sequential building blocks," *ACM SIGOPS/EuroSys European Conference on Computer Systems*, 2007.
[4] Y. Yu, M. Isard, D. Fetterly, M. Budiu, U. Erlingsson, P. K. Gunda, and J. Currey, "DryadLINQ: A system for general-purpose distributed data-parallel computing using a high-level language," *USENIX conference on Operating Systems Design and Implementation*, 2008.
[5] X. Huang and A. Madan, "CAP3: A DNA sequence assembly program," *Genome Research*, vol. 9, p. 868, 1999.
[6] M. A. Batzer and P. L. Deininger, "Alu repeats and human genomic diversity," *Nature Reviews Genetics*, vol. 3, pp. 370-379, 2002.
[7] C. Cortes, Support-vector network, 1995.
[8] F. Perronnin, and C. Dance, "Fisher Kernels on Visual Vocabularies for Image Categorization," IEEE Conference on Computer Vision and Pattern Recognition, 2007.
[9] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," IEEE Conference on Computer Vision and Pattern Recognition, 2010.
[10] C. P. Shen, C. H. Liu, F. S. Lin, H. Lin, C. Y. F. Huang, C. Y. Kao, F. Lai, J. W. Lin, "A Multiclass Classification Tool Using Cloud Computing Architecture," *International Symposium on Network Enabled Health Informatics, Biomedicine and Bioinformatics* (*HI-BI-BI 2012*), Istanbul, Turkey, June. 2012. pp. 797-802.
[11] http://en.wikipedia.org/wiki/Apache_Hadoop
[12] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, and J. P. Mesirov, "Multiclass cancer diagnosis using tumor gene expression signatures," Proceedings of the National Academy of Sciences, vol. 98, pp. 15149, 2001.
[13] A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy, "A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis," Bioinformatics, vol. 21, pp. 631-643, 2005.

TABLE I. TABLE I: CLASSIFICATION RESULTS*

	Total	BL	BR	CNS	CO	LE	LU	LY	ME	ML	OV	PA	PR	RE	UT	Accuracy(%)
BL	**3**	**3**														**100**
BR	**4**		**4**													**100**
CNS	**4**			**4**												**100**
CO	**4**				**4**											**100**
LE	**6**					**6**										**100**
LU	**4**						**3**				**1**					**75**
LY	**6**							**6**								**100**
ME	**3**								**3**							**100**
ML	**2**									**2**						**100**
OV	**4**										**4**					**100**
PA	**3**											**3**				**100**
PR	**6**												**6**			**100**
RE	**3**													**3**		**100**
UT	**2**														**2**	**100**
Tota	**54**	**3**	**4**	**4**	**4**	**6**	**3**	**6**	**3**	**2**	**5**	**3**	**6**	**3**	**2**	**98.1**

* **BL**, bladder transitional cell carcinoma; **BR**, breast adenocarcinoma; **CNS**, central nervous system; **CO**, colorectal adenocarcinoma; **LE**, leukemia; **LU**, lung adenocarcinoma; **LY**, lymphoma; **ME**, pleural mesothelioma; **ML**, melanoma; **OV**, ovarian adenocarcinoma; **PA**, pancreatic adenocarcinoma; **PR**, prostate adenocarcinoma; **RE**, renal cell carcinoma; **UT**, uterine adenocarcinoma.