Multiclass classification of microarray data samples with Flexible Neural Tree

Xuejiao Lei
School of Information science and Engineering, University of Jinan
Jinan, PR China
aduosi@126.com

Yuehui Chen
School of Information science and Engineering, University of Jinan
Jinan, PR China
yhchen@ujn.edu.cn

Abstract—A reliable and precise classification of tumors is essential for successful diagnosis and treatment of cancer. But microarray data are often extremely asymmetric in dimensionality, such as thousands or even tens of thousands of genes and a few hundreds of samples. This paper proposes the multiclass Flexible Neural Tree (FNT) algorithm for cancer classification. Based on the pre-defined instruction/operator sets, a flexible neural tree model can be created and evolved. The FNT structure is developed using probabilistic incremental program evolution (PIPE) and the free parameters embedded in the neural tree are optimized by particle swarm optimization (PSO) algorithm. Empirical results on two well-known cancer datasets show competitive results with other methods.

Keywords-microarray; Flexible Neural Tree; Probabilistic Incremental Program Evolution; Particle Swarm Optimization

I. INTRODUCTION

Microarray technology enables the recording of expression levels of a number of genes simultaneously for various cancerous tissues samples. The classification of tissues based on the gene expression profiles presents a major signification to cancer diagnosis and drug discovery. An accurate prediction of cancer has great value in proving better treatment and response to therapy varying from different aspects [1].

In recent years, various artificial intelligence methods have been applied to enhance the diagnoses procedures and to aid the physician's efforts. However, the microarray data generally has tens of thousands of genes and dozens of samples. For such a dimension space, it is extremely difficult to use traditional classification methods directly. So gene selection methods have been proposed and developed to reduce the dimensionality. There are some gene selection methods for multiclass microarray data, such as chi-squared, Information Gain. Along with the feature selection methods, intelligent methods have been applied for microarray classification: support vector machine (SVM)[2], K nearest neighbor (KNN)[3], artificial neural network (ANN)[4] and so on. But high accurate classification is difficult to achieve. Most intelligent classifiers are apt to be over-fitted. Also, most of the proposed methods can't be directly applied to multiclass classification, for example SVM.

Multiclass classification techniques can be roughly divided into two types. One type is the binary classification algorithms that can be naturally extended to handle multiclass problems directly. The other type is the decompositions of multiclass problems into binary ones. While binary classification problems are the simplest of all, but many real-world problems are multiclass problems. Simultaneously, Scholkopf and Smola note that there is probably no multiclass method that outperforms everything else and that for practical purposes the choice of the method has to be made depending on the constraints, such as the desired level of accuracy, the time available for development and training, and the nature of the classification problems [5].

Therefore, in this paper, we adopt the second technique that is decompositions of multiclass problems into binary ones. We utilize the pairwise comparison classification strategy[6]. A new method——the flexible neural tree (FNT) is employed to be the base classifier. The FNT structure is developed using probabilistic incremental program evolution (PIPE) and the free parameters embedded in the neural tree are optimized by particle swarm optimization (PSO) algorithm.

The main steps of this experiment are listed as follows: firstly pre-propose the cancer gene expression datasets, secondly extract important features, thirdly build up models to predict and analyze the result finally.

II. GENE SELECTION METHOD

Generally, the microarray data has very high dimensionality genes (in thousands) and small size of samples (in dozens). However, only small parts of genes have great impact on classification and most of them are useless. These irrelevant genes not only confuse learning algorithms, but also degrade their performance and efficiency. Moreover, the prediction model induced from irrelevant genes may prone to over-fitting. In addition, reducing the number of genes can help to cut down the inputs for computation, so the classifiers are much more efficient for classification and run much faster. For this so-called "high-dimensional small sample" problem, a suitable gene

selection method is very important. Bscatter employed in this paper is introduced as follows.

Bscatter[7] is a correlation score for feature ranking to handle multi-class classification score. In another word, it is a ratio between-class scatter to within-class scatter. How to get the value of Bscatter? We describe below:

For each class i and each feature j, we define:

$$\mu_{j,i} = \frac{1}{|C_i|} \sum_{X \in C_i} x_j \tag{1}$$

$\mu_{j,i}$ represents the mean value of feature j for class C_i. We also define the total mean along feature j:

$$\mu_j = \frac{1}{m} \sum_{X} x_j \tag{2}$$

Using equations (1) and (2), we provide a measure of the between-class scatter along feature j:

$$B_j = \sum_{i=1}^{C} |C_i| (\mu_{j,i} - \mu_j)^2 \tag{3}$$

This leads to the following score function:

$$\text{BScatter}_j = B_j \Big/ \sum_{i=1}^{C} \sigma_{ji} \qquad j = 1, \ldots\ldots, n \tag{4}$$

where σ_{ji} is the standard deviation of class i along feature j.

This score is related to Fisher discrimination analysis for multiple classes [8] under feature independence assumption. It credits the largest score to the feature that maximizes the ratio of the between-class scatter to the within-class scatter. The highest Bscatter_j value is most informative and the expression levels differ most on average in the classes while also favoring those with small deviation in the respective classes. Then the genes with high Bscatter_j values are selected as the top features.

III. MULTICLASS FLEXIBLE NEURAL TREE

In this paper, we use a completely new method—flexible neural tree (FNT) [9] as the base classifier and the pairwise comparison classification strategy. Depending on the extracted datasets, the FNT model can be created and evolved. The structure of FNT is developed by the Probabilistic Incremental Program Evolution (PIPE) and the parameters are optimized by the Particle Swarm Optimization (PSO).

The framework of FNT allows input variables selection, over-layer connections and different activation functions for the various nodes involved [10]. The FNT model has greatly improved the performance of optimizing and designing neural network structure. It can automatically design and find the better network's structure and parameters according to the input features. By the evolutionary algorithm, the individuals of FNT tend to simplify structure of the similar model accuracy. Also the final structure of FNT is generally simpler than that of neural network, but has better generalization ability; FNT can automatically select the input variables or features that contribute more to the model accuracy.

A) Flexible Neural Tree

FNT model is comprised of the function set F and terminal instruction set T used for generating a FNT model described as follows:

$$S = F \cup T = \{+2, +3, \ldots\ldots, +N\} \cup \{x_1, x_2 \ldots\ldots x_n\} \tag{5}$$

The F set +i(i = 2, 3, . . ., N) are non-leaf nodes' instructions described as flexible neuron operators which has i inputs. And x1, x2,. . ., xn, are leaf nodes' instructions, i.e., terminal arguments. The output of a flexible neuron operator is calculated as following flexible activation function.

$$out_n = f(a_n, b_n, net_n) = e^{-(net_n - a_n / b_n)^2} \tag{6}$$

$$net_n = \sum_{j=1}^{n} w_j \times x_j \tag{7}$$

In the neural tree's creation process, if +i(i = 2, 3, 4, . . ., N) is selected, i real values which are denoted as W are randomly generated and used for representing the connection strength between the node +i and its children. In addition, two adjustable activation parameters a_i and bi are randomly created. Fig 1 is a typical flexible neuron operator and a FNT tree model.

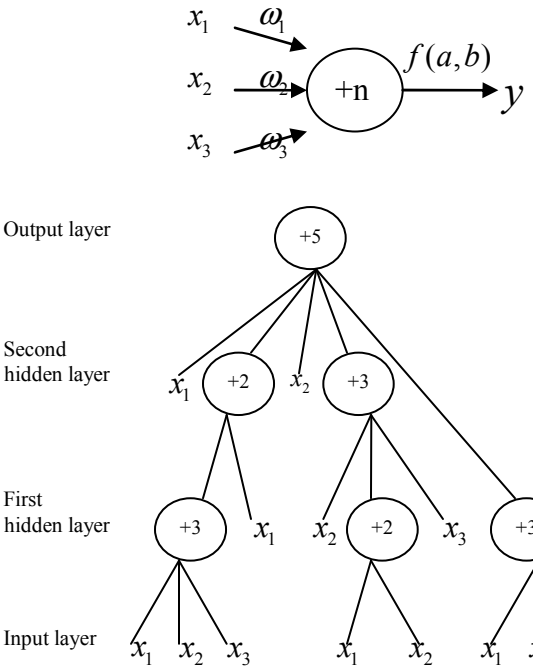

Figure 1 A flexible neuron operator (up) and a typical FNT model (down).

B) Tree structure optimization with PIPE

In the probabilistic incremental program evolution (PIPE) algorithm [11] computer programs or mathematical expressions are evolved like in GP [12]. PIPE programs are encoded in n-ary trees that are parsed depth first from left to right, with n being the maximal number of function arguments. PIPE generates programs according to a probability distribution over all possible programs composable from the instruction set ($F \cup T$). The probability distribution is stored in an underlying probabilistic

prototype tree (PPT). The PPT contains at each node an initial probability for each instruction from F U T and a random constant from [0,1). Programs are generated by traversing the PPT depth first starting at the root node. At each node an instruction is picked according to the node's probability distribution. In case the generic random constant is picked it is instantiated either to the value stored in the PPT node or a random value from [0,1), depending on the instruction's probability. To adapt PPTs probabilities PIPE generates successive populations of programs. It evaluates each program of a population and assigns it a scalar, non-negative "fitness value", which reflects the program's performance. PIPE adapts PPTs probabilities so that the overall probability of creating the best program of the current population increases. Finally PPT's probabilities are mutated to better explore the search space.

C) Parameter optimization with PSO

The Particle Swarm Optimization[13] conducts searches using a population of particles which correspond to individuals in evolutionary algorithm (EA). A population of particles is randomly generated initially. Each particle represents a potential solution and has a position represented by a position vector x_i. A swarm of particles moves through the problem space, with the moving velocity of each particle represented by a velocity vector v_i. At each time step, a function f_i representing a quality measure is calculated by using x_i as input. Each particle keeps track of its own best position, which is associated with the best fitness it has achieved so far in a vector p_i. Furthermore, the best position among all the particles obtained so far in the population is kept track of as p_g. In addition to this global version, another version of PSO keeps track of the best position among all the topological neighbors of a particle. At each time step t, by using the individual best position, p_i, and the global best position, $p_g(t)$, a new velocity for particle i is updated by:

$$v_i(t+1) = v_i(t) + c_1\theta_1(p_i(t) - x_i(t)) + c_2\theta_2(p_g(t) - x_i(t))$$

where c_1 and c_2 are positive constant and θ_1 and θ_2 are uniformly distributed random number in [0, 1]. The term v_i is limited to the range of $\pm v_{\max}$. If the velocity violates this limit, it is set to its proper limit. Changing velocity this way enables the particle i to search around its individual best position, p_i, and global best position, p_g. Based on the updated velocities, each particle changes its position according to the following equation $x_i(t+1) = x_i(t) + v_i(t+1)$.

D) Procedure of the General Learning Algorithm

The general learning procedure for constructing the FNT model can be described as follows.

1) Create an initial population randomly (FNT trees and its corresponding parameters);
2) Structure optimization is achieved by using the PIPE algorithm;
3) If a better structure is found, then go to step 4), otherwise go to step 2);
4) Parameter optimization is achieved by the PSO algorithm as described in subsection III.C. In this stage, the architecture of FNT model is fixed, and it is the best tree developed during the end of run of the structure search. The parameters (weights and flexible activation function parameters) encoded in the best tree formulate a particle.
5) If the maximum number of local search is reached, or no better parameter vector is found for a significantly long time then go to step 6); otherwise go to step 4);
6) If satisfactory solution is found, then the algorithm is stopped; otherwise go to step 2).

IV. CANCER CLASSIFICATION USING FNT

A) Data sets

We performed extensive experiments on two benchmark cancer datasets, namely the MLL, Lymphoma.

- MLL dataset

This dataset consists of gene expression profiles of three classes of leukemia and is available at http://research.nhgri.nih.gov/micr-oarray/Supplement. It was first studied by Scott et al. [14] in proposing that a distinct disease type, MLL, can be clearly separated from conventional acute lymphoblastic leukemia (ALL) and acute myelogenous leukemias (AML).The dataset includes 12582 probe sets from the Affymetrix chip, and contains 72 samples. The numbers of samples in the three classes are balanced, 24 in ALL, 20 in MLL, and 28 in MLL.

- The Lymphoma dataset

We used the Lymphoma dataset of Alizadeh et al. [15]. This dataset is a broad term encompassing three cancers of the lymphatic system and is available at http://www.stat.cmu.ed-u/~jiashun/Research/software/Data/Lymphoma/. This dataset con-tains 4026 genes and 62 samples, where the quantities of patients with these three types of lymphoma are 11 in Chronic Lymphocytic, 42 in Follicular and 9 in Diffuse Large B-cell Lymphoma.

The normalization procedure is firstly used for preprocessing the raw data. Four steps were taken:

1) If a value is greater than the floor 16,000 and smaller than the ceiling 100, this value is replaced by the ceiling/floor.
2) Leaving out the genes with (max - min)≤500, here max and min refer to the maximum and minimum of the expression values of a gene, respectively.
3) Carrying out logarithmic transformation with 10 as the base to all the expression values.
4) For each gene i, subtract the mean measurement of the gene μ_i and divide by the standard deviation σ_i.

After this transformation, the mean of each gene will be zero, and the standard deviation will be one.

After this steps, the feature selection method(in section II) is then employed to form the feature subsets and re-sampling method is used to form the training and testing datasets.

In our experiment, we utilize the Bscatter gene selection method selecting 10 informative genes and the pairwise comparison strategy for multiclass classification, the FNT is employed to be the base classifier. We use PSO to adjust the parameter of each FNT and PIPE to optimize the structure of each FNT.

The classification performance was measured using 5-fold cross validation technique. That is the gene expression vectors are randomly partitioned into 5 equally sized subsets, and each subset is used as a test set for a classifier trained on the remaining 4 subsets. The training data is used to select informative features. This process is repeated 10 times to obtain the average results with experiments in total.

A comparison of different classification methods for MLL and Lymphoma datasets are shown in Table 1 and Table 2. For the two dataset, our method produces the best classification, the average accuracy of MLL is 98.6%, and the average accuracy of Lymphoma reach to 100%. Also, at first, with the same number of informative genes, but after the training, the number of genes which are used for the SVM, Radial Basis Functions(RBF), Naïve Bayes network (BBN) classifiers is invariant, whereas with the FNT classifier, the number of informative genes on MLL and Lymphoma reduce to 9 and 8.

TABLE 1 RELATIVE WORKS ON MLL DATASET

method	Test accuracy (%)
Our method	**98.6**
SVM[7]	**94.4**

TABLE 2 RELATIVE WORKS ON LYMPHOMA DATASET

method	Test accuracy (%)
Our method	**100**
BBN[16]	**92**
RBF[16]	**98**

V. CONCLUSION

In this paper, the multiclass flexible neural tree models are used for cancer classification. The MLL and Lymphoma datasets are used for conducting all the experiments. Gene features are first extracted by the correlation analysis technique which greatly reduces dimensionality as well as maintains the informative features. Then the FNTs are employed to classify. Compare the results with some advanced artificial techniques; the proposed method produces the best recognition rates. The next work, we could abstract more critical features to characterize the gene expression data. We also believe that FNT can have a great contribution to this subject.

ACKNOWLEDGMENTS

This research was partially supported by the Natural Science Foundation of China (61070130, 60903176, 60873089), the Program for New Century Excellent Talents in university (NCET-10-0863), the Natural Science Foundation of Shandong Province, China (ZR2010FQ020), the Shandong Distinguished Middle-aged and Young Scientist Encourage and Reward Foundation, China (BS2009SW003), the China Postdoctoral Science Foundation (20100470081), and the Shandong Provincial Key Laboratory of Network Based Intelligent Computing.

References

[1] Y. Lu, J. Han, Cancer classification using gene expression data. Information Systems. 28 (2003) 243-268

[2] Alireza Osareh, Bita Shadgar. Microarray Data Analysis for Cancer Classification. IEEE (2009) 978-1-4244-5969-8/10

[3] M.B. Eisen, B.O. Brown, DNA arrays for analysis of gene expression, Methods Enzymol. 303 (1999) 179–205.

[4] Alexander Stantnikov, Constantin F. Aliferis et al. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics (2005) 631-643

[5] B. Scholkopf and J. Smola Learning with Kernels. MIT Press, Cambridge, MA. 2002

[6] Kreeel,U.H.-G.Pairwise classification and support vector machines. In Scholkopf,B., Burges,C. and Smola,A.J. (eds), Advances in Kernel Methods—Support Vector Learning, The MIT Press, Cambridge, MA (1999), pp. 255–268.

[7] Hong Chai and Carlotta Domeniconi. An Evaluation of Gene Selection Methods for Multi-class Microarray Data Classification. Proceedings of the Second European Workshop on Data Mining and Text Mining in Bioinformatics.

[8] Duda, R. O. , Hart, P., Pattern classification and scene analysis, Wiley, 1973.

[9] Y. Chen, B. Yang, J. Dong, A. Abraham. Time-series forecasting using flexible neural tree model, Inf. Sci. 174 (2005) 219–235.

[10] Yuehui Chen, Bin Yang, Qingfang Meng. Small-time scale network traffic prediction based on flexible neural tree [J]. Applied Soft Computing 12 (2012) 274–279.

[11] R. Salustowiczand J. Schmidhuber. Probabilistic incremental program evolu-tion. Evolutionary Computation, 5(2):123-141, 1997.

[12] KAROLCHIK D, BAERTSCH R, DIEKHANS M, et al. The UCSC genome browser database [J].Nucleic Acids Res, 2003, 31(1):51–4.

[13] KENNEDY J , EBERHART R. Particle swarm optimization[C]. Proceedings of the 1995 IEEE International Conference on Neural Networks,1995:1942−1948

[14] Scott, A., Armstrong, et al., MLL Translocations Specify A Distinct Gene Expression Profile that Distinguishes A Unique Leukemia. Nature Genetics, 30:41-47, January 2002.

[15] Alizadeh A., Eisen M., et al. Distinct types of diffuse large b-cell-lymphoma identified by gene expression profiling, Nature 403 (2000) 503–511.

[16] Nikita V. Orlov, Wayne W. chen, et al. Automatic Classification of Lymphoma Images With Transform-Based Global Features. IEEE (2010) 1089-7771