Gene Expression Profiling Using Flexible Neural Trees

Yuehui Chen[1], Lizhi Peng[1], and Ajith Abraham[1,2]

[1] School of Information Science and Engineering
Jinan University, Jinan 250022, P.R. China
yhchen@ujn.edu.cn
[2] IITA Professorship Program, School of Computer Science and Engg.
Chung-Ang University, Seoul, Republic of Korea
ajith.abraham@ieee.org

Abstract. This paper proposes a Flexible Neural Tree (FNT) model for informative gene selection and gene expression profiles classification. Based on the pre-defined instruction/operator sets, a flexible neural tree model can be created and evolved. This framework allows input variables selection, over-layer connections and different activation functions for the various nodes involved. The FNT structure is developed using the Extended Compact Genetic Programming and the free parameters embedded in the neural tree are optimized by particle swarm optimization algorithm. Empirical results on two well-known cancer datasets shows competitive results with existing methods.

1 Introduction

The classification of cancers from gene expression profiles is actively investigated in bioinformatics. It commonly consists of feature selection and pattern classification. In advance, feature selection selects informative features useful to categorize a sample into predefined classes from lots of gene expression profiles. Pattern classification is composed of learning a classifier with those features and categorizing samples with the classifier.

Much research effort has been devoted to exploring the informative gene selection from microarray data. Typical effective feature reduction methods include principal components analysis (PCA), class-separability measure, Fisher ratio and t-test. Evolutionary based feature selection methods are alternatives of the gene selection approaches. A probabilistic model building genetic algorithm based informative selection method was proposed in [1]. Genetic programming can be also used to select informative gene and classification of gene expression profiles [2]. After the gene selection was performed, many candidate classifiers can be employed for classification of microarray data, including Bayessian network, KNN, neural networks, support vector machine [12], random forest [4] etc.. For a recent review, the reader is refer to ref. [3]. Classification algorithms that directly provide measures of variable importance are of great interest for gene selection, specially if the classification algorithm itself presents features that make

E. Corchado et al. (Eds.): IDEAL 2006, LNCS 4224, pp. 1121–1128, 2006.

it well suited for the types of problems frequently faced with microarray data. Random forest is one such algorithm [4]. The proposed FNT method is another alternative.

This papers proposes a Flexible Neural Tree (FNT) [5][6] for selecting the input variables and forecasting exchange rates. Based on the pre-defined instruction/operator sets, a flexible neural tree model can be created and evolved. FNT allows input variables selection, over-layer connections and different activation functions for different nodes. In our previous work, the hierarchical structure was evolved using Probabilistic Incremental Program Evolution algorithm (PIPE) with specific instructions. In this research, the hierarchical structure is evolved using the Extended Compact Genetic Programming (ECGP), a tree-structure based evolutionary algorithm. The fine tuning of the parameters encoded in the structure is accomplished using particle swarm optimization (PSO). The proposed method interleaves both optimizations. Starting with random structures and corresponding parameters, it first tries to improve the structure and then as soon as an improved structure is found, it fine tunes its parameters. It then goes back to improving the structure again and, fine tunes the structure and rules' parameters. This loop continues until a satisfactory solution is found or a time limit is reached. The novelty of this paper is in the usage of flexible neural tree model for selecting the informative genes and for classification of microarray data.

2 The Flexible Neural Tree Model

The function set F and terminal instruction set T used for generating a FNT model are described as $S = F \bigcup T = \{+_2, +_3, \ldots, +_N\} \bigcup \{x_1, \ldots, x_n\}$, where $+_i (i = 2, 3, \ldots, N)$ denote non-leaf nodes' instructions and taking i arguments. x_1,x_2,. . .,x_n are leaf nodes' instructions and taking no other arguments. The output of a non-leaf node is calculated as a flexible neuron model (see Fig.1). From this point of view, the instruction $+_i$ is also called a flexible neuron operator with i inputs.

In the creation process of neural tree, if a nonterminal instruction, i.e., $+_i (i = 2, 3, 4, \ldots, N)$ is selected, i real values are randomly generated and used for representing the connection strength between the node $+_i$ and its children. In addition, two adjustable parameters a_i and b_i are randomly created as flexible activation function parameters. For developing the forecasting model, the flexible activation function $f(a_i, b_i, x) = e^{-(\frac{x-a_i}{b_i})^2}$ is used. The total excitation of $+_n$ is $net_n = \sum_{j=1}^{n} w_j * x_j$, where $x_j (j = 1, 2, \ldots, n)$ are the inputs to node $+_n$. The output of the node $+_n$ is then calculated by $out_n = f(a_n, b_n, net_n) = e^{-(\frac{net_n - a_n}{b_n})^2}$. The overall output of flexible neural tree can be computed from left to right by depth-first method, recursively.

2.1 Tree Structure Optimization

Finding an optimal or near-optimal neural tree is formulated as a product of evolution. In our previous studies, the Genetic Programming (GP) and Probabilistic

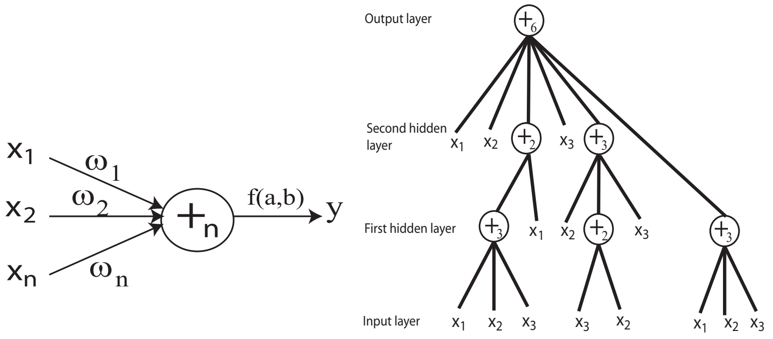

Fig. 1. A flexible neuron operator (left), and a typical representation of the FNT with function instruction set $F = \{+_2, +_3, +_4, +_5, +_6\}$, and terminal instruction set $T = \{x_1, x_2, x_3\}$ (right)

Incremental Program Evolution (PIPE) have been explored for structure optimization of the FNT [5][6]. In this paper, the Extended Compact Genetic Programming (ECGP) [7] is employed to find an optimal or near-optimal FNT structure.

ECGP is a direct extension of ECGA to the tree representation which is based on the PIPE prototype tree. In ECGA, Marginal Product Models (MPMs) are used to model the interaction among genes, represented as random variables, given a population of Genetic Algorithm individuals. MPMs are represented as measures of marginal distributions on partitions of random variables. ECGP is based on the PIPE prototype tree, and thus each node in the prototype tree is a random variable. ECGP decomposes or partitions the prototype tree into sub-trees, and the MPM factorises the joint probability of all nodes of the prototype tree, to a product of marginal distributions on a partition of its sub-trees. A greedy search heuristic is used to find an optimal MPM mode under the framework of minimum encoding inference. ECGP can represent the probability distribution for more than one node at a time. Thus, it extends PIPE in that the interactions among multiple nodes are considered.

2.2 Parameter Optimization with PSO

The Particle Swarm Optimization (PSO) conducts searches using a population of particles which correspond to individuals in evolutionary algorithm (EA) [9]. A population of particles is randomly generated initially. Each particle represents a potential solution and has a position represented by a position vector $\mathbf{x_i}$. A swarm of particles moves through the problem space, with the moving velocity of each particle represented by a velocity vector $\mathbf{v_i}$. At each time step, a function f_i representing a quality measure is calculated by using $\mathbf{x_i}$ 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 $\mathbf{p_i}$. Furthermore, the best position among all the particles obtained so far in the population is kept track of as $\mathbf{p_g}$. 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, $\mathbf{p_i}$, and the global best position, $\mathbf{p_g(t)}$, a new velocity for particle i is updated by

$$\mathbf{v_i(t+1)} = \mathbf{v_i(t)} + c_1\phi_1(\mathbf{p_i(t)} - \mathbf{x_i(t)}) + c_2\phi_2(\mathbf{p_g(t)} - \mathbf{x_i(t)}) \tag{1}$$

where c_1 and c_2 are positive constant and ϕ_1 and ϕ_2 are uniformly distributed random number in [0,1]. The term $\mathbf{v_i}$ is limited to the range of $\pm\mathbf{v_{max}}$. 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, $\mathbf{p_i}$, and global best position, $\mathbf{p_g}$. Based on the updated velocities, each particle changes its position according to the following equation:

$$\mathbf{x_i(t+1)} = \mathbf{x_i(t)} + \mathbf{v_i(t+1)}. \tag{2}$$

2.3 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 ECGP 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 2. 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).

2.4 Feature/Input Selection Using FNT

It is often a difficult task to select important variables for a forecasting or classification problem, especially when the feature space is large. A fully connected NN classifier usually cannot do this. In the perspective of FNT framework, the nature of model construction procedure allows the FNT to identify important input features in building a forecasting model that is computationally efficient and effective. The mechanisms of input selection in the FNT constructing procedure are as follows. (1) Initially the input variables are selected to formulate the FNT model with same probabilities; (2) The variables which have more contribution to the objective function will be enhanced and have high opportunity to survive in the next generation by a evolutionary procedure; (3) The evolutionary operators i.e., crossover and mutation, provide a input selection method by which the FNT should select appropriate variables automatically.

3 Cancer Classification Using FNT Paradigms

3.1 Data Sets

The colon cancer dataset contains gene expression information extracted from DNA microarrays [1]. The dataset consists of 62 samples in which 22 are normal samples and 40 are cancer tissue samples, each having 2000 features. We randomly choose 31 samples for training set and the remaining 31 samples were used as testing set. (http://sdmc.lit.org.sg/GEDatasets/Data/ColonTumor.zip). The leukemia dataset consists of 72 samples divided into two classes ALL and AML [14]. There are 47 ALL and 25 AML samples and each contains 7129 features. This dataset was divided into a training set with 38 samples (27 ALL and 11 AML) and a testing set with 34 samples (20 ALL and 14 AML) (Availble at: http://sdmc.lit.org.sgGEDatasets DataALL-AML_Leukemia.zip).

3.2 Colon Cancer

The data was randomly divided into a training set of 30 samples and testing set of 12 for 50 times, and our final results were averaged over these 30 independent trials (Fig. 2). A FNT model was constructed using the training data and then the model was used on the test data set. The instruction sets used to create an optimal FNT forecaster is $S = F \bigcup T = \{+_5, +_6, \ldots, +_9\} \bigcup \{x_0, x_1, \ldots, x_{1999}\}$. Where $x_i (i = 0, 1, \ldots, 1999)$ denotes the 2000 input variables (genes) of the classification model.

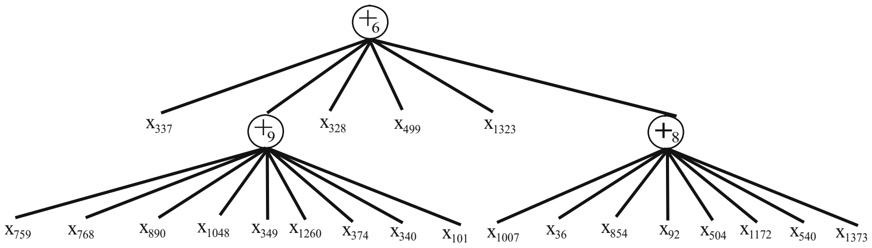

Fig. 2. An evolved best FNT for colon data classification

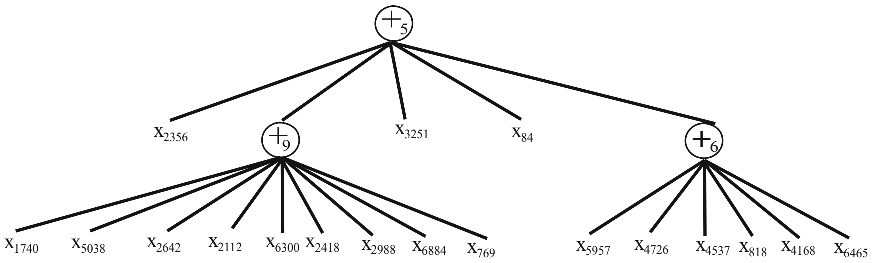

Fig. 3. An evolved best FNT for leukemia data classification

Table 1. The extracted informative genes in case of Colon dataset

$x_{337}, x_{328}, x_{759}, x_{768}, x_{890}, x_{1048}, x_{349}, x_{1260}, x_{374}, x_{340}, x_{101}, x_{499}, x_{1007}, x_{36}, x_{854}, x_{92}, x_{504}, x_{1172}, x_{540}, x_{1373}, x_{1323}$

Table 2. The extracted informative genes in case of leukemia dataset

$x_{2356}, x_{3251}, x_{1740}, x_{5038}, x_{2642}, x_{2112}, x_{6300}, x_{2418}, x_{2988}, x_{6884}, x_{769}, x_{5957}, x_{4726}, x_{4537}, x_{818}, x_{4168}, x_{6465}, x_{84}$

Table 3. The best prediction rate of some studies in case of Colon dataset

Classifier	Classification rate (%)
GA+SVM [10]	84.7± 9.1
Bootstrapped GA+SVM [11]	80.0
Combined kernel for SVM [12]	75.33±7.0
FNT (This paper)	97.09±0.018

Table 4. The best prediction rate of some studies in case of Colon dataset

Classifier	Classification rate (%)
Weighted voting [8]	94.1
Bootstrapped GA+SVM [11]	97.0
Combined kernel for SVM [12]	85.3±3.0
Multi-domain gating network [13]	75.0
FNT (This paper)	99.6±0.021

A best FNT tree obtained by the proposed method is shown in Figure 2. It should be noted that the important features for constructing the FNT model were formulated in accordance with the procedure mentioned in the previous section. These informative genes selected by FNT algorithm is shown in Table 1.

For comparison purpose, the classification performances of a genetic algorithm trained SVM [10], Bootstrapped GA+SVM [11], Combined kernel for SVM [12] and the FNT method proposed in this paper are shown in Tables 3. It is observed that the proposed FNT classification models are better than other models for classification of microarray dataset.

3.3 Leukemia Cancer

As mentioned in Sec. 3.1, the Leukemia dataset is already divided into training and testing set. To setup the 30 independent trials, A FNT model was constructed using the training data and then the model was used on the test data set. The instruction sets used to create an optimal FNT forecaster is $S = F \bigcup T = \{+_5, +_6, \ldots, +_9\} \bigcup \{x_0, x_1, \ldots, x_{7128}\}$. Where $x_i (i = 0, 1, \ldots, 7128)$ denotes the 7129 input variables (genes) of the classification model.

A best FNT tree obtained by the proposed method for leukemia cancer classification is shown in Figure 3. It should be noted that the important features for constructing the FNT model were formulated in accordance with the procedure mentioned in the previous section. These informative genes selected by FNT algorithm is shown in Table 2.

For comparison purposes, the classification performances of Weighted voting method [8], Bootstrapped GA+SVM [11], Combined kernel for SVM [12], Multi-domain gating network [13] and the FNT method proposed in this paper are shown in Table 4. It is observed that the proposed FNT classification models are better than other models for classification of microarray dataset.

4 Conclusions

In this paper, we presented a Flexible Neural Tree (FNT) model for informative gene selection and classification of microarray data simultaneously. We have demonstrated that the FNT classification model may provide better classifier than the other classification models. The experimental results also shown a significantly improvement in classification accuracy compare to other classifiers especially in case of Leukemia cancer dataset. This implies that the proposed FNT model can be used as a feasible solution for classification of microarray data.

Acknowledgment

This research was partially supported the Natural Science Foundation of China under contract number 60573065, and The Provincial Science and Technology Development Program of Shandong under contract number SDSP2004-0720-03.

References

1. Topon, K. P. and Hitoshi, I.: Gene Selection for Classification of Cancers using Probabilistic Model Building Genetic Algorithm. BioSystems 82(3)(2005) 208-225.
2. Hong, J.-H. and Cho, S.-B.: The classification of cancer based on DNA microarray data that uses diverse ensemble genetic programming. Artificial Intelligence in Medicine, 36 (2006) 43-58
3. Asyali, M.H., Colak, D., Demirkaya, O., Inan, M. S.: Gene Expression Profile Classification: A Review. Current Bioinformatics. 1 (2006) 55-73.
4. Ramn Daz-Uriarte and Sara Alvarez de Andrs: Gene selection and classification of microarray data using random forest. BMC Bioinformatics. 7 (2006) 3.
5. Chen, Y., Yang, B. and Dong, J.: Nonlinear System Modeling via Optimal Design of Neural Trees. International Journal of Neural Systems. 14 (2004) 125-137

6. Chen, Y., Yang, B., Dong, J. and Abraham, A.: Time-series Forecasting using Flexible Neural Tree Model. Information Science. 174 (2005) 219-235
7. Sastry, K. and Goldberg, D. E.: Probabilistic model building and competent genetic programming. In: R. L. Riolo and B. Worzel, editors, Genetic Programming Theory and Practise. (2003) 205-220
8. Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, J. P., Mesirov, J., Coller, H., Loh, M. L., Downing, J.R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E.: Mo-lecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, Science, vol. 286 (1999): 531-537.
9. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. Proc. IEEE Int. Conf. on Neural Networks, Perth, (1995) 1492-1948.
10. Frohlich, H., Chapelle, O., and Scholkopf, B.: Feature Selection for Support Vector Ma-chines by Means of Genetic Algorithms, 15th IEEE International Conference on Tools with Artificial Intelligence (2003): 142
11. Chen, Xue-wen: Gene Selection for Cancer Classification Using Bootstrapped Genetic Algorithms and Support Vector Machines, IEEE Computer Society Bioinformatics Confer-ence (2003): 504
12. Nguyen, H.-N, Ohn, S.-Y, Park, J., and Park, K.-S.: Combined Kernel Function Approach in SVM for Diagnosis of Cancer, Proceedings of the First International Conference on Natural Computation (2005)
13. Su, T., Basu, M., Toure, A.: Multi-Domain Gating Network for Classification of Cancer Cells using Gene Expression Data, Proceedings of the International Joint Conference on Neural Networks (2002): 286-289
14. Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., and Levine, A.: Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays, Proceedings of National Academy of Sciences of the United States of American, 96 (1999) : 6745-6750.