Combined Genetic Programming for Microarray Data Possible Biomarkers for cancer Data

[1] Jayaker J, [2]Ms M.Dhivya

[1] M Tech Information Technology, [2] Assistance Professor
Department of Information Technology, Faculty of Engineering and Technology, SRM University, Chennai, India
[1] jayaker.j@gmail.com , [2] dhivya.ma@ktr.srmuniv.ac.in

***Abstract*— Researchers have found different types of cancer cell along with various normal gene structures in Microarray data. It is possible to set benchmark for finding out affected cell from normal one using various machine learning technique. Due to wide range of gene about thousand of them and minimum training data there occurs imbalance between them. This difference can be minimized using various optimizing algorithm and machine learning technique. In this paper we proposed Combined Genetic Programming for Microarray Data along with Majority Voting(MV) for classification. Genetic program along with MV act as both classifier and gene selection. The potential challenge for genetic program is it has to find gene type and also has to find optimal solution.**

***Index Terms*— Preprocessing, Genetic Program(GP), majority voting(MV), analyzing frequently affected gene**)

I. Introduction

Microarray technology is the recent advancement in medical field which helps in creating a large database holding both cancerous tissue and normal tissue. Researcher have found that there is big difference in gene expression level of normal tissue and tumor tissue. They are trying to create pattern with these difference in gene expression using several machine learning technique to find out behavior of cancer. riske involved in this clinical behavior is correlating the small training sample with large number of gene. Mostly used method for finding out these gene expression along with microarray data is clustering, k-nearest neighbor(KNN) classifier, support vector machine(SVM) and naïve-Bayes classifier. Genetic algorithm(GA) is mostly used method around world for gene selection, some other classification algorithm under (GA) is random probabilistic model building genetic algorthim(RPMBGA), probabilistic model building genetic algorthim(PMBGA) has been used for this over fitting of gene with cancer data.

Recently genetic programming is technique which is replacing the old computation method like genetic algorthim, because it act as both classifier as well as gene selection algorthim. Patient sample or training data set of gene expression is given to genetic programming along with their defined parameter at intial stage, then GP evolve us with binary or boolen expression of gene describing which class the genes for the given training samples.

Even though the result obtained in the GP is potential challenging due to their large number hence therefore another computational solution is required to optimize the solution. To overcome this problem we come up with majority voting technique along with gene classifier. In this we develop several rule which can be more accurate than single rule or member of group. We count all their votes in favor of particular class. Then the assigned samples is segregated to that class and highest number of votes are calculated for given class

II. Methods

In this we discuss about our steps required for classification of gene expression data

2.1 Preprocessing

Microarray data file contain gene chip software which holdes all normal gene and cancer affected gene in it. Each file contain column row, where each column contain expression level of different gene in single sample and row contains expression level of single gene in different sample. In this certain value may be negative, some may have high threshold value, some may have lower threshold value. preprocessing is mainly done to eliminate background correction, log transformation, normalization, summarization.

2.1.1 Background correction

This is method of removing unwanted data's (+,'-'ve) values presents in data sets. Unwanted positive and negative value which highly not recommended in the given data set should be removed using any preprocessing method.

2.1.2 Log transformation

Converting the data set to fit particularly into GP on the basis of threshold value. Preprocessing helps in squeezing larger value and stretching smaller value to help in meeting the assumption.

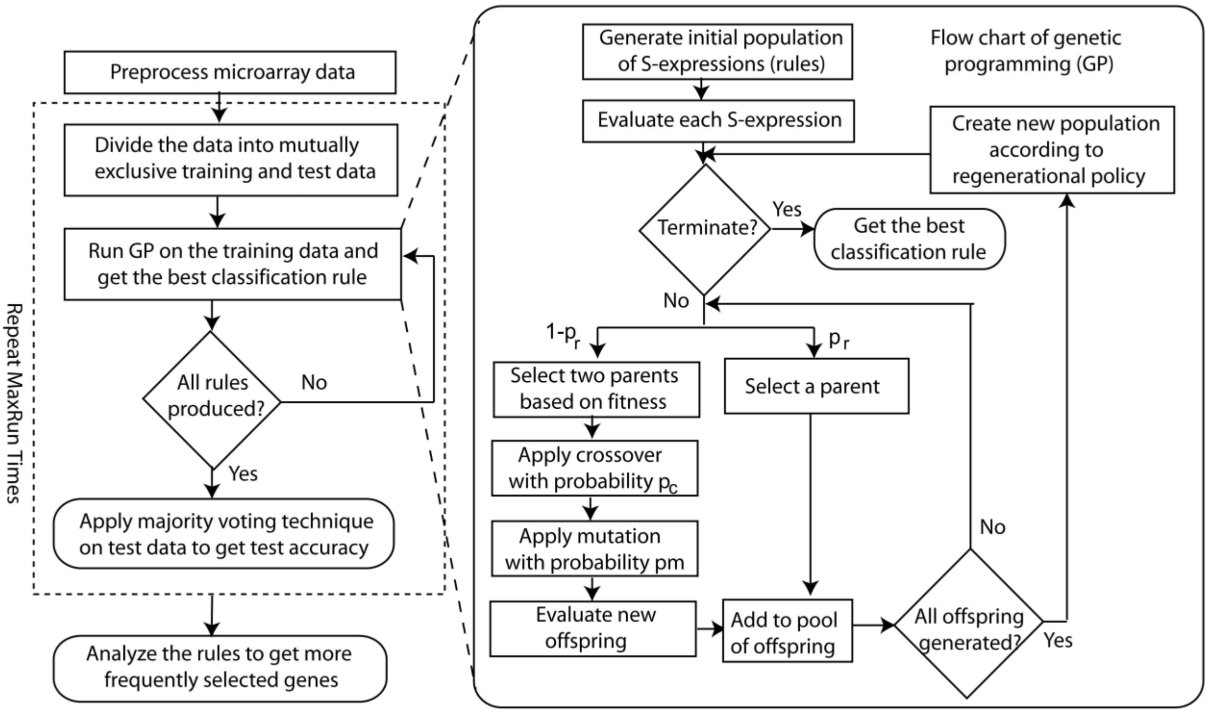
2.1.3 Normalization

Process of reducing unwanted variation in data redundant helps in better optimization of data set inside GP hence normalized value should be obtained using preprocessing technique.

2.1.4 Summarization

Summarization is method of normalizing gene value to fit into array. This helps in overall organization of different classes of data.

Preprocessing can be done with some other third party tool like weka which provide user preprocessing of larger data on the bases of classifier, clustering, associate algorithm. This can be run in both consol mode or in GUI mode. It also helps in providing visual representation or graph based reduction of dataset.


Preprocess microarray data
Divide the data into mutually exclusive training and test data
Run GP on the training data and get the best classification rule
All rules produced?
No
Yes
Apply majority voting technique on test data to get test accuracy
Repeat MaxRun Times
Analyze the rules to get more frequently selected genes
Generate initial population of S-expressions (rules)
Flow chart of genetic programming (GP)
Evaluate each S-expression
Create new population according to regenerational policy
Terminate?
Yes
Get the best classification rule
No
1-p_r
p_r
Select two parents based on fitness
Select a parent
Apply crossover with probability p_c
Apply mutation with probability pm
Evaluate new offspring
Add to pool of offspring
All offspring generated?
No
Yes

III. Genetic Programming

Genetic programming is the automated lraning of computer program. GP's learning algorithm is inspired by the theory of evolution and our contemporary understanding of biology and natural evolution. In GA there is fixed length of variable it will be in the form of binary but in GP its of variable length of tree with functions and variables hence crossover and mutation operator are applied.

1. Initial we have to generate initial population of random composition of function and terminal set.
2. Fitness is calculated from the individual in the population on training samples.
3. When the termination criteria are not met, do the following steps

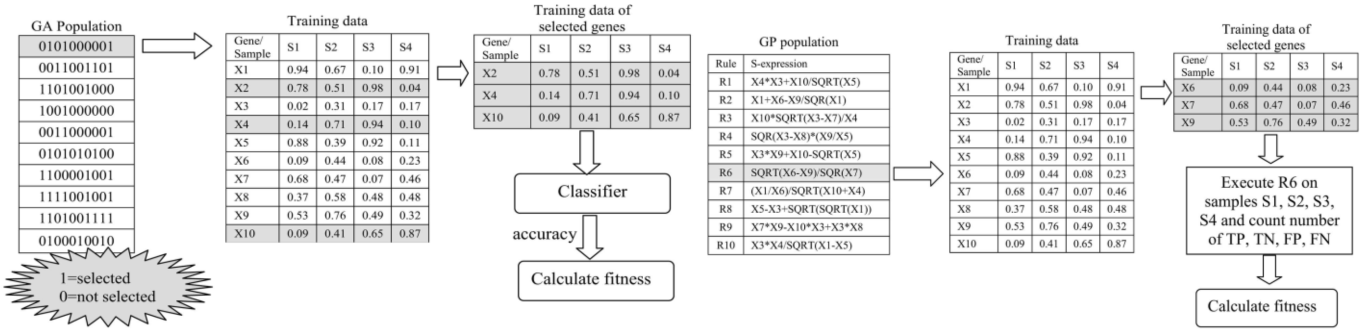
Create new offspring from the parents that are selected from population based on fitness.

1. Reproduction: Copy selected parents to the new population without any modification.

2. Crossover: The crossover operator combines the genetically material of two parents by swapping a part of one parents with part of other parents.

3. Mutation: It operates on only one individual. Normally after crossover has occurred, each child produced by the crossover undergoes mutation with a low probability.

Factors like population size, crossover depth, crossover probability, reproduction probability are associated with Genetic program. The above steps helps in determining the initial population from GP and create reduced data set for the given one.



GA Population

GA Population
0101000001
0011001101
1101001000
1001000000
0011000001
0101010100
1100001001
1111001001
1101001111
0100010010

1=selected
0=not selected

Training data

Gene/ Sample	S1	S2	S3	S4
X1	0.94	0.67	0.10	0.91
X2	0.78	0.51	0.98	0.04
X3	0.02	0.31	0.17	0.17
X4	0.14	0.71	0.94	0.10
X5	0.88	0.39	0.92	0.11
X6	0.09	0.44	0.08	0.23
X7	0.68	0.47	0.07	0.46
X8	0.37	0.58	0.48	0.48
X9	0.53	0.76	0.49	0.32
X10	0.09	0.41	0.65	0.87

Training data of selected genes

Gene/ Sample	S1	S2	S3	S4
X2	0.78	0.51	0.98	0.04
X4	0.14	0.71	0.94	0.10
X10	0.09	0.41	0.65	0.87

GP population

Rule	S-expression
R1	X4*X3+X10/SQRT(X5)
R2	X1+X6-X9/SQR(X1)
R3	X10*SQRT(X3-X7)/X4
R4	SQR(X3-X8)*(X9/X5)
R5	X3*X9+X10-SQRT(X5)
R6	SQRT(X6-X9)/SQR(X7)
R7	(X1/X6)/SQRT(X10+X4)
R8	X5-X3+SQRT(SQRT(X1))
R9	X7*X9-X10*X3+X3*X8
R10	X3*X4/SQRT(X1-X5)

Training data

Gene/ Sample	S1	S2	S3	S4
X1	0.94	0.67	0.10	0.91
X2	0.78	0.51	0.98	0.04
X3	0.02	0.31	0.17	0.17
X4	0.14	0.71	0.94	0.10
X5	0.88	0.39	0.92	0.11
X6	0.09	0.44	0.08	0.23
X7	0.68	0.47	0.07	0.46
X8	0.37	0.58	0.48	0.48
X9	0.53	0.76	0.49	0.32
X10	0.09	0.41	0.65	0.87

Training data of selected genes

Gene/ Sample	S1	S2	S3	S4
X6	0.09	0.44	0.08	0.23
X7	0.68	0.47	0.07	0.46
X9	0.53	0.76	0.49	0.32

Execute R6 on samples S1, S2, S3, S4 and count number of TP, TN, FP, FN
Calculate fitness

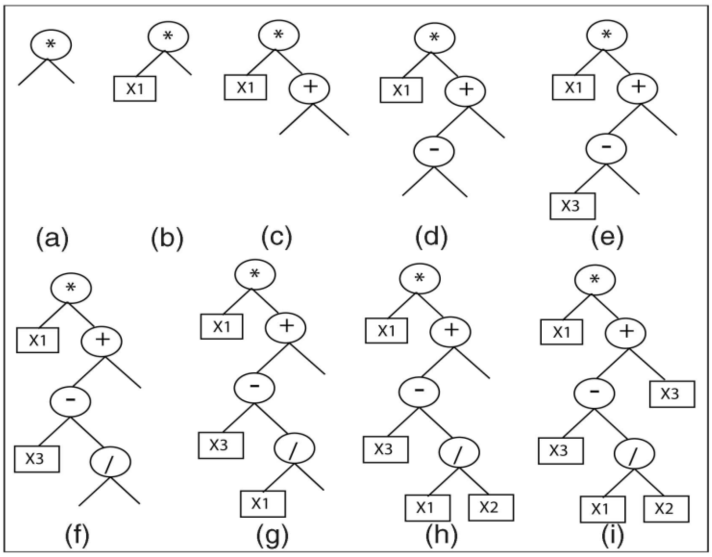
In GA algorithm initial population is created on the basis of binary rule. Rule s assigned as 0,1 and training data's are introduced in GP these training data are processed based on these 0's and 1 rule selected gene are established. Then the obtained data are invoked in separate GA algorithm for classifier, ie in what class the gene are distributed. Based on the result fitness are calculated for the given data set this process of shrinking of bigger dataset in GA to smaller one involves two to three algorithm.

In GP initial population is created on the basis of set of s-expression rule on training data. These training data flows into these rule and genes are selected from these training data. Then each rule is deployed on the extracted data based on the sample and true positive value true negative, false positive false negative are calculated. Then fitness is calculated from the above rule.

Generation of intial population

Intial population is generated by selecting random terminal with restriction of size. The pseudocode for

```
 GenerateRule(Tree t, Integer depth)
If ðdepth < 1Þ Then return;
ElseIf ðdepth ¼ 1Þ Then
t.value=SelectTerminalRandomly();
t.left=null; t.right=null; return;
Else
node=SelectNodeRandomly();
If (node is a terminal) Then
t.value=node; t.left=null;
t.right=null; return;
ElseIf (node is a unary function) Then
t.value=node; t.right=null;
t.left=new Tree();
GenerateRule(t.left,depth-1);
Else
t.value=node;
t.left=new Tree(); t.right=new Tree();
GenerateRule(t.left,depth-1);
GenerateRule(t.right,depth-1);
```



Creation of s-expression of maximum depth with operator { + , _, *, /,sqr} and operand{x1,x2,x3} first multiplication factor is randomly chosen because its binary function which requires two operand. X1is assigned to its left and "+" as second argument. This continue tell all function in the tree get their argument.

IV. MAJORITY VOTING

Even though GP produced a desired output for the given dataset but it couldn't fit with large number of gene with very small training data. So in order to over come this majority voting is used along with GP. Leave one out cross validation (LOOCV) method is used, where one sample is left out from training sample. LOOCV helps in overall classification, based on number of samples per training data. Majority voting can be explained by following example if we want to know the labels of A and B. We run GP for X times to get X best rule, if the result produced from S-expression is positive we favor the vote in class A, if its in negative we favor the vote in class B finally the total number of votes are calculated and leading vote class is used at the end. This is applicable only for single class.

For multiclass majority voting is used differently, if there are C class of sample in dataset we generate X*C rules in GP. We use all sample of class i which is evolved during the rule for class I as positive and others as negative hence its is used as binary classifier. If the result of S-expression is positive we favor our vote in positive class of i, if its negative we favor our vote in negative class of i.

V. RESULTS

Brain cancer data

Brain cancer dataset contains 12625 genes which is divided into two subset of classic and non classic gliomas. Brain cancer dataset are available in http://www genome.wi.mit. edu/cancer/pub/glioma. These dataset are subjected to preprocessing with standard lower threshold=20, higher threshold=16000 and difference=5. Then preprocessed dataset are divided into training set and test subset in the ratio 2:1. These data are used in GP to find its accuracy and test result.

Prostate cancer data

Prostate cancer dataset contains 12600 gene which is divided from 52 prostate tumor and 50 nonprostate tumor. These dataset are available in http://www.genome.wi.mit.edu/MPR/prostate. Data set is subjected to preprocessing with standard lower threshold=10, higher threshold=16000, difference=50. These preprocessed data are divided into training set and test subset in the ratio 1:1. These data are used in GP to find its accuracy and test result.

Breast cancer data

Breast cancer dataset contains 5361 genes and they are broadly classified as BRCA1 and BRCA2. Dataset for breast cancer is available in http:// research.nhgri.nih.gov/microarray/NEJM_Supplement/. These dataset is subjected to preprocessing with standard lower threshold =20, higher threshold=16000 and differernce=5. These preprocessed data are divided into training set and test subset in the ratio 2:1. Once preprocessing is finished dataset are subjected to GP to find its accuracy and test result.

Lung carcinoma data

Lung carcinoma dataset contains 12600 they have various internal classification. Dataset for lung carcinoma is available in http://research.dfci.harvard.edu/ meyersonlab/lungca.html. These dataset is subjected to preprocessing and they are divided into training set and test subject in the ratio 1:1. Once preprocessing is finished its subject to GP to find its accuracy and test result.

Test accuracy on dataset

Four public cancer data are subjecte to experiment with different number of rules. Intial population is created with random seeds on each time GP runs. Minimum number of member in a voting group should be three to take any decision. For binary classification we used brain cancer dataset and prostate cancer dataset and for multiclass classification we used breast cancer and lung carcinoma with v=5. We performed v=3c where c is the number of class in dataset.

GP along with majority voting have produced overall accuracy of 94.12 on different dataset and could able to select frequently occurring gene in these dataset.

VI. CONCLUSION

In this paper, we propose genetic programming along with majority voting technique for predicting cancer class along with various test data and test sample. Four publicly available dataset are been used in this paper to test its accuracy and class prediction. Accuracy obtained in multiple rule or multiple set of rules which we used in majority voting is far more accurate than single rule or single set of rule.

Some unresolved issues still exist in this that is how to find out quantitative relationship exist among the frequently obtained genes and how majority voting along with GP can handle other larger multiclass dataset

REFERENCES

[1] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine, “Broad Patterns of Gene Expression Revealed by Clustering of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays,” Proc. Nat’l Academy Science USA,

[2] A. Alizadeh, M. Eisen, R. Davis, C. Ma, I. Lossos, A. Rosenwald, J. Boldrick, H. Sabet, T. Tran, X. Yu, J. Powell, L. Yang, G. Marti, T. Moore, J.J. Hudson, L. Lu, D. Lewis, R. Tibshirani, G. Sherlock, W. Chan, T. Greiner, D. Weisenburger, J. Armitage, R. Warnke, R. Levy, W. Wilson, M. Grever, J. Byrd, D. Botstein, P. Brown, and L. Staudt, “Distinct Types of Diffuse Large B-Cell Lymphoma Identified by Gene Expression Profiling,” Nature, vol. 403, no. 6781, pp. 503-511, 2000.

[3] A. Ben-Dor, R. Shamir, and Z. Yakhini, “Clustering Gene Expression Patterns,” J. Computational Biology, vol. 6, pp. 281-297, 1999.

[4] M.B. Eisen, P.T. Spellman, P. Brown, and D. Botstein, “Cluster Analysis and Display of Genome-Wide Expression Patterns,” Proc. Nat’l Academy Sciences USA, vol. 95, pp. 14 863-14 868, 1998.

[5] A. Bhattacharjee, W. Richards, J. Stauton, C. Li, S. Monti, P. Vasa,C. Ladd, J. Behesti, R. Buneo, M. Gillete, M. Loda, G. Weber, E. Mark, E. Lander, W. Wong, B. Johnson, T. Golub, D. Sugarbaker, and M. Meyerson, “Classification of Human Lung Carcinomas by mRNA Expression Profiling Reveals Distinct Adenocarcinoma Subclasses,” Proc. Nat’l Academy Science USA, vol. 98, pp. 13 790- 13 795, 2001.

[6] C. Nutt, D. Mani, R. Betensky, P. Tamayo, J. Cairncross, C. Ladd, U. Pohl, C. Hartmann, M. McLaughlin, T.T. Batchelor, P. Black, A. von Deimling, S. Pomeroy, T. Golub, and D. Louis, “Gene Expression-Based Classification of Malignant Gliomas Correlates Better with Survival than Histological Classification,” Cancer Research, vol. 63, no. 7, pp. 1602-1607, 2003.

[7] D. Singh, P. Febbo, K. Ross, D. Jackson, J. Manola, C. Ladd, P. Tamayo, A. Renshaw, A. D'Amico, J. Richie, E. Lander, M. Loda, P. Kantoff, T. Golub, and W. Sellers, "Gene Expression Correlates of Clinical Prostate Cancer Behavior," CancerCell,http://www.cancercell.org/cgi/content/full/1/2/203, Mar. 2002.

[8] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," Science, vol. 286, no. 15, pp. 531-537, 1999.

[9] C.H.Q. Ding, "Unsupervised Feature Selection via g in Gene Expression Analysis," Bioinformatics, vol. 19, no. 10, pp. 1259-1266, 2003.

[10] P. Park, M. Pagano, and M. Bonnetti, "A Nonparametric Scoring Algorithm for Identifying Informative Genes from Microarray Data," Proc. Pacific Symp. Bioinformatics (PSB '01), vol. 6, pp. 30-41, 2001.