Int J Pharm Bio Sci 2015 April; 6(2): (B) 663 - 667

Research Article **Allied Science**

International Journal of Pharma and Bio Sciences **ISSN 0975-6299**

GENE SELECTION USING BACTERIAL FORAGING OPTIMIZATION

SUNITA BENIWAL* AND DHARMINDER KUMAR

Department of Computer Science and Engineering, Guru Jambeshwar University of Science and Technology.Hisar, Haryana

ABSTRACT

Microarray gene expression data can be analyzed for diagnosis of cancer and its stage. It usually concerns a very large number of variables relative to a small number of observations. This makes application of data mining techniques difficult and so to reduce the data dimensionality some pre-processing technique needs to be used. In this paper dataset used for analysis is about lung cancer consisting of 96 samples. Bacterial foraging optimization algorithm has been used in this paper for selecting relevant genes.The bacterial foraging optimization algorithm is an optimization technique which derives its idea from for aging behavior of Bacteria *E. coli*.It shows good performance by selecting less and relevant genes.

KEYWORDS: Microarray, Bacterial Foraging Optimization, Support vector machines.

*Corresponding author

SUNITA BENIWAL
Department of Computer Science and Engineering, Guru Jambeshwar University of Science and Technology.Hisar, Haryana

INTRODUCTION

Data mining can be used on medical data for disease diagnosis, identification of subtypes of disease and sometimes to mark the level of illness. Different classification techniques can be used to distinguish an ill person from a healthy person and to differentiate between disease subtypes. Microarray data of patients suffering from different disease causing mutation in one or more than one gene is available widely. A model can be built using available data and that model can then be used for diagnosis of disease in new patients. Mining techniques can be used on microarray data to find out the affected genes and for diagnosis of type of disease. DNA microarray offers the ability to measure levels of expressions of thousands of genes simultaneously. The hypothesis that many or all human diseases may be accompanied by specific changes in gene expression has generated much interest among the bioinformatics community and can be used in classification of patient samples based on gene expression. The classification of cancers from gene expression profiles is an active research area in bioinformatics. This paper focuses on classification of lung cancer patients data with the help of Support Vector machine. Before applying support vector machines (SVM), data was preprocessed using bacterial foraging optimization algorithm (BFOA). The paper is organised as follows: first of all microarray data and problems encountered in mining such data is briefly explained.BFOAand SVM techniques are also explained followed by implementation details and results and conclusion.

BACKGROUND

DNA microarray offers the ability to measure levels of expressions of thousands of genes simultaneously. It facilitates the prognosis and discovering of subtypes of disease. For using microarray technology for diagnosis an accurate method to extract knowledge and useful information from these microarray gene expression datasets is required. Microarray data that are currently available concern a very large number of variables relative to a small number of observations. This characteristic, known as the curse of dimensionality, makes application of different data mining methods difficult and so to reduce the data dimensionality some pre-processing technique needs to be used for selection of most informative genes. In this paper BFO algorithm was used for preprocessing. Dataset used in this work is of lung cancer. It contains data regarding 96 persons. Out of 96, 86 persons are cancer patients, remaining is the data of normal healthy persons. Data regarding 7129 genes is present initially, which is first preprocessed and then classified.

Bacterial foraging optimization

The bacterial foraging optimization algorithm proposed by Passino [1] is an optimization technique which derives its idea from foraging behavior of Bacteria *E. coli*. The *E. coli*bacteria grows at a very fast rate if given suitable condition and sufficient food to grow. The *E. coli* bacteria moves very fast into nutrient areas and tries to go away from noxious substances. The motions of bacteria are known as taxes. In foraging theory, the objective is to search for and obtain nutrients in a fashion that energy intake per unit time (E/T) is minimized [8].An *E.coli*bacterium has 8-10 flagella placed randomly on its body with a speed of 100-200 rps. Its movement and behavior is characterized by the spinning flagella which acts as a Biological motor and helps bacteria to swim. The bacterial foraging process consists mainly of four sequential mechanisms, namely chemotaxis, swarming, reproduction and elimination-dispersal which are briefly explained below:

1. Chemotaxis

Bacteria have a tendency to gather to the nutrient-rich areas by an activity called Chemotaxis. An *E.coli* bacterium can move in two different ways: it can run (swim for a period of time) or tumble and alternates between these movements throughout its travel in search of food. In BFO, a unit walk with random direction represents a “tumble” and a unit walk with the

same direction in the last step indicates "run". The flagella can rotate either clockwise or counterclockwise. When all the flagella rotate counterclockwise, they form a compact, helically propelling the cell along a trajectory, which is called "run". When the flagella rotate in clockwise direction they enable the bacterium to move in different directions and cause bacteria to "tumble".A bacterium $\theta_i(j, k, l)$ represents a potential solution to the optimization problem, where iis the bacterium in its jth chemotactic loop, its kth reproduction loop, and its lth elimination-dispersal loop. Within the chemotactic loop, a bacterium tumble was modeled by Passino [1] with a random search direction φ(i) as presented in Equation 1:

$\varphi(i) = \Delta(i) \_\Delta(i)T \Delta(i)$ (1),

where Δ(i) is a randomly generated vector of size n with elements within the following interval: [–1, 1]. Afterwards,each bacterium ichanges its position by a swim movement as indicated in Equation 2:

$\theta_i(j + 1, k, l) = \theta_i(j, k, l) + C(i)\varphi(i)$ (2)

where $\theta_i(j + 1, k, l)$ is the new position of bacterium i, $\theta_i(j, k, l)$ is the current position of bacterium i, C(i) is thestepsize value defined by the user, and φ(i) is the tumble value as detailed previously in Equation 1. The swim will be repeated Ns times if and only if the new position is better than the previous one, i.e., (assuming minimization) $f(\theta_i(j + 1, k, l)) < f(\theta_i(j, k, l))$. Otherwise, a new tumble is computed. The chemotactic loop stops whenthe chemotactic loop limit Ncis reached for all bacteria in the swarm.

2. Swarming

A bacterium in times of stresses releases attractants to signal the bacteria to swarm together. Each bacterium also releases repellant to signal the others to be at a minimum distance from it. Thus all of them will have a cell to cell attraction via attractant and cell to cell repulsion via repellant. This process requires the definition of a set of parameter values by the user[1].

3. Reproduction

After the completion of all Nc chemotactic steps a reproduction step takes place. Fitness value of the bacteria is stored in ascending order and eliminating half of them with the worst value. The remaining half will be duplicated so as to maintain a fixed swarm size.

4. Elimination and Dispersal

To eliminate the probability of bacteria being stuck around the initial or local optima positions, the bacteria are diversified either gradually or suddenly and global optima is obtained. The dispersion operation takes place after a certain number of reproduction steps. A bacterium is chosen, according to a present probability Ped , to be dispersed and moved to another position within the environment. This may disturb optimization process but prevent the local minima trapping.

BFO has been used by many researchers for optimization. BFOA and Particle Swarm Optimization (PSO) together for tuning a Fractional order speed controller in a Permanent Magnet Synchronous Motor Drive performed well as compared to that of PSO, BFO and the gradient descent method. [2] Kim et al. [3]proposed a hybrid approach involving genetic algorithms (GA) and bacterial foraging (BF) algorithms which was used to tune a PID controller of an automatic voltage regulator (AVR) showing the proposed approach to be very efficient. Bacterial foraging optimization (BFO) has also been used to train neural networks showing better speed and accuracy than Genetic algorithms and neural networks[4].

Support vector machines

Support Vector Machines [5] are basically binary classification algorithms derived from statistical learning theory. They have been applied successfully in fields such as text categorisation, hand-written character recognition, image classification, biosequences analysis, etc. The SVM separates the classes with a decision surface that maximizes the margin between the classes. The surface is often called the optimal hyper plane, and the

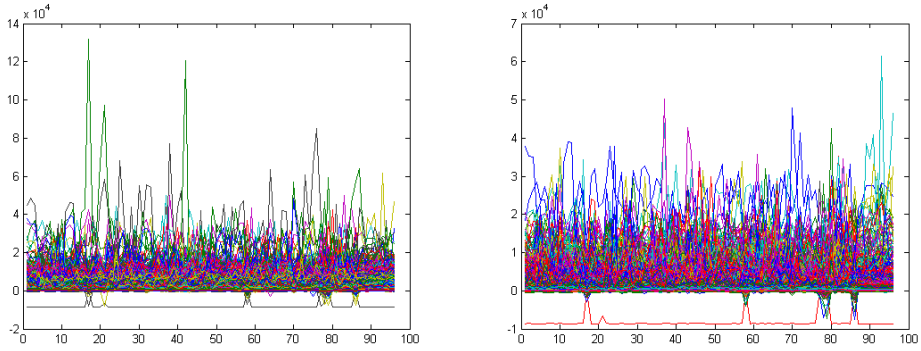
data points closest to the hyper plane are called support vectors. The support vectors are the critical elements of the training set. The mechanism that defines the mapping process is called the kernel function. The SVM can be adapted to become a nonlinear classifier through the use of nonlinear kernels. SVM can function as a multiclass classifier by combining several binary SVM classifiers. The output of SVM classification is the decision values of each pixel for each class, which are used for probability estimates. The probability values represent "true" probability in the sense that each probability falls in the range of 0 to 1, and the sum of these values for each pixel equals 1. Classification is then performed by selecting then highest probability. SVM includes a penalty parameter that allows a certain degree of misclassification, which is particularly important for non-separable training sets. The penalty parameter controls the trade-off between allowing training errors and forcing rigid margins. It creates a soft margin that permits some misclassifications, such as it allows some training points on the wrong side of the hyperplane. Increasing the value of the penalty parameter increases the cost of misclassifying points and forces the creation of a more accurate model that may not generalize well [6].

RESULTS AND DISCUSSION

All experiments were carried out using Matlab.Dataset used in this paper iscomposed of persons suffering from lung cancer and healthy persons. Dataset has 96 samples and each sample has 7129 genes.Before using the dataset for designing a classifier, dataset is preprocessed. Preprocessing was performed using filter selection approach. Genes are first selected and then only they are used for classification.BFO algorithm has been used to remove the irrelevant genes. BFO algorithm was run with parameters like number of chemo-tactic steps (Nc), number of reproduction steps (Nre), number of elimination and dispersal steps (Ned), dispersal probability (Ped), number of bacteria (N) & swim length (Ns). Attractant and repellant value were also initialized.The whole process is run Ned number of times, which is number of elimination and dispersal events. In each elimination and dispersal events, reproduction is done Nre times and in each elimination-dispersal event chemotaxis is done. Whether a gene is to swim or tumble is decided by the value of fitness function as shown by eq. 1.

fposition= abs(mean(x(ClassA,:))-mean(x(ClassB,:))./sqrt(var(x(ClassA,:)) ./ nA + var(x(ClassB,:)) ./ nB) eq. 1

hereclassA and classB represents two classes of lung cancer dataset used in this paper i.e. normal and cancerous samples.nAis number of samplesin class A. nB represents no of samples in class B.After calculating fitness of each bacterium, bacteria which are headed in right direction swim and bacteria headed in wrong direction tumble. Bacteria are arranged in order according to their nutrient concentration. The least fit bacteria do not reproduce and most fit bacteria are split into two identical copies, so keeping only most fit bacteria in next generation. And after executing the whole process ned number of times i.e. running the procedure for assigned number of elimination and dispersal events, only most fittest bacteria are retained. If we talk in terms of genes, only the genes having more relevance to lung cancer are retained. All other genes which are the same in cancerous and normal persons are removed. The number of genes selected in this process is 2931 genes.



a) b)

Figure 1
Distribution of dataset
a) before preprocessing b) after preprocessing

Figure 1 shows the distribution of dataset used in the paper. Figure 1 (a) shows the original dataset distribution and it is evident that outliers are also present. After the BFO algorithm is applied for gene selection, some genes are removed and only relevant genes are left. Figure 1(b) shows the distribution after dataset is preprocessed.After selecting the relevant genes any classification technique can be applied to distinguish cancerous samples from normal samples. As only relevant genes are selected, the performance of classifier in terms of time taken and accuracy may be improved by using the selected genes. In this paper a very simple fitness function has been used for BFOA. The fitness function used by the BFO can be improved so as to select lesser number of genes and select more relevant genes. Also BFOA can be combined with other statistical techniques to give better results. The performance of BFOA can be tested on other datasets.

REFERENCES

1. Passino K. Biomimicry of bacterial foraging for distributed optimizationand control. IEEE Control Systems Magazine, 22(3), 52–67. (2002).
2. Anguluri R., Abraham A., Snasel V. A Hybrid Bacterial Foraging - PSO Algorithm Based Tuning of Optimal FOPI Speed Controller. ActaMontanisticaSlovacaRoˇcník16(1), 55-65. (2011).
3. Kim DH., Abraham A., Cho JH. A hybrid genetic algorithm and bacterial foraging approach for global optimization. Information Sciences, 177, 3918–3937. (2007).
4. Zhang Y., Wu L., Wang S. Bacterial Foraging Optimization Based Neural Network for Short-term Load Forecasting.Journal of Computational Information Systems 6(7), 2099-2105. (2010).
5. Vapnik VN, Ed. Statistical Learning Theory, Vol 2, Wiley, New York. 401-430, (1998).
6. Hsu, Chih-Wei., Chang, Chih-Chung and Lin, Chih-Jen. “A practicalguide to support vector classification. Accessed on “26 November 2014. “http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.(2003).