Genomics 107 (2016) 231–238

Contents lists available at ScienceDirect

Genomics

journal homepage: www.elsevier.com/locate/ygeno

A hybrid gene selection approach for microarray data classification using cellular learning automata and ant colony optimization

Fatemeh Vafaee Sharbaf [a], Sara Mosafer [a], Mohammad Hossein Moattar [b,*]

[a] Department of Computer Engineering, Imam Reza International University, Mashhad, Iran
[b] Department of Software Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

ARTICLE INFO

Article history:
Received 2 January 2016
Received in revised form 20 April 2016
Accepted 1 May 2016
Available online 3 May 2016

Keywords:
Gene selection
Microarray data
Cellular learning automata
Ant colony optimization
K-nearest neighbor
Naïve Bayes

ABSTRACT

This paper proposes an approach for gene selection in microarray data. The proposed approach consists of a primary filter approach using Fisher criterion which reduces the initial genes and hence the search space and time complexity. Then, a wrapper approach which is based on cellular learning automata (CLA) optimized with ant colony method (ACO) is used to find the set of features which improve the classification accuracy. CLA is applied due to its capability to learn and model complicated relationships. The selected features from the last phase are evaluated using ROC curve and the most effective while smallest feature subset is determined. The classifiers which are evaluated in the proposed framework are K-nearest neighbor; support vector machine and naïve Bayes. The proposed approach is evaluated on 4 microarray datasets. The evaluations confirm that the proposed approach can find the smallest subset of genes while approaching the maximum accuracy.

1. Introduction

Analyzing the gene expression level, one can gather valuable information regarding the mutual influence of genes in a genetic network [1].In the databases used for gene expression analysis, the number of samples is few but the dimension is too high. These two factors make classification and data analysis challenging. However, all genes do not participate in the occurrence of cancer. Using all genes to discriminate and classify cancer may lead to incorrect decisions. Using feature selection techniques to identify the effective subset of features is an important issue in the problem of gene expression analysis. The main goal of feature selection is to identify a minimum subset of features that increase the decision accuracy.

Traditional feature selection approaches are divided into four categories namely filter, wrapper, embedded and hybrid approaches. In filter approaches each feature is evaluated individually [2]. These approaches can be easily applied to high dimensional datasets; their complexity is low and the approaches are classifier independent. For this purpose, measures such as t-test [3], information gain [4], minimum Redundancy Maximum Relevance (mRMR) [5] and Euclidean distance [6] are the most popular. In this type of feature selection approaches, the features which have the best statistical score are selected. In filter feature selection approaches, the performance of the classifier and interdependency of the features play no role, therefore it is not surprising that the performance of the classifier would be low or redundant features may be found in the selected feature set [7].

In wrapper approaches, classifier performance is used as the measures for feature evaluation. Wrapper approaches are categorized as deterministic and stochastic approaches. Sequential forward selection (SFS) and Sequential backward elimination (SBE) are categorize as deterministic and optimization based approaches such as randomized hill climbing [8], Ant colony [9] and genetic algorithms [10] are stochastic approaches. Although the classifier performance is high for this approaches but the search space complexity is very high for the problems with thousands of feature and this leads to higher time complexity.

Embedded approaches take advantage of the model properties to analysis the problem and select the most important features [11]. Approaches such as decision tree and neural network fall in this group of methods, however these approaches are also of high computational complexity. Guyon et. al. [12] introduced one of the most widely applied embedded techniques based on support vector machine and Recursive Feature Elimination (SVM-RFE) for gene selection and cancer classification. Also, Maldonado et. al. [13] proposed an embedded approach by introducing a penalty factor in the dual formulation of SVM.

None of the above mentioned approaches are able to overcome all the problems solely. Therefore ensemble approaches are proposed in the literature [14,15]. In these approaches, feature selection is done using a hybrid model and the results are integrated. Mundra et al. hybridize two of the most popular feature selection approaches, namely SVM-RFE and mRMR [16]. Shreem et. al. [17] proposed RM-GA approach

[*] Corresponding author.
E-mail addresses: vafaeeshaarbaf@gmail.com (F. Vafaee Sharbaf), sa.mosafer90@yahoo.com (S. Mosafer), moattar@mshdiau.ac.ir (M.H. Moattar).

http://dx.doi.org/10.1016/j.ygeno.2016.05.001
0888-7543/© 2016 Elsevier Inc. All rights reserved.

which was a hybrid of ReliefF, mRMR and genetic algorithm (GA). Chuang et al. [18] proposed a hybrid approach named CFS-TGA which was the hybrid of correlation based feature selection (CFS) and Taguchi-Genetic Algorithm (TGA) and used KNN as the classifier. Lee and Liu [19] proposed an approach called Genetic Algorithm Dynamic Parameter (GADP) for producing every possible subset of genes and rank the genes using their occurrence frequency. Also, Yassi and Moattar [20] proposed a feature selection approach for microarray data which combined both ranking methods and wrapper approaches to satisfy the data scarcity problem.

In this paper, we have proposed an ensemble approach to select the smallest subset of features to have the best possible classifier performance. This approach consists of two phases. The first phase uses a filter and the second phase is based on a wrapper approach. In the first phase, the features are ranked using Fisher criterion. The use of the filter approach is intended to lower the search space complexity. Then, the best features from the previous stage are fed to the wrapper approach which is based on the hybrid cellular learning automata and ant colony optimization. The rest of this paper is organized as follows. Section 2 introduces the main materials of the proposed approach including cellular automata and ant colony optimization. Section 3 explains the proposed methodology. The evaluation datasets are described in Section 4. Section 5 summaries the experimental results and discussions. Finally conclusions and guide for feature works are offered in Section 6.

2. Materials and methods

2.1. Cellular learning automata

Cellular learning automata (CLA) are system modeling approaches which consists of simple basic parts. In CLA, the behavior of every part is modified based on the behaviors of its neighbors and its personal previous experiments. The simple parts of this model can show complicated functionalities via interactions with each other. A CLA is a cellular automaton in which every cell (or a group of cells) is equipped with learning capability.

Local rule, φ controls the cellular automata and determines if a selected action should be punished or rewarded. The rewards and punishes leads to the structural update of the cellular learning automata to achieve a specific objective. A cellular learning automaton is denoted by a penury $<\Lambda, A, \Omega, \varphi, L>$. $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ denotes the set of cells in the cellular learning automata which constructs a Cartesian network. $A = \{a_1, a_2, \ldots, a_k\}$ is the set of allowed actions of a CLA in a cell. $A^t(\lambda_i)$ denotes the executed action in time t and cell λ_i and φ is the rule with governs the cellular learning automata. Ω is the neighboring cells and L is the set of learning cells. Depending on the application, the neighboring cells are determined using different approaches (i.e. Von Neumaan, Smith, Moore and Cole neighborhood) [21]. Learning automata is capable of simulating complicated systems using simple interactions of cells, and hence is appropriate for solving NP-complete problems.

2.2. Ant colony optimization

Ant colony optimization is a meta-heuristic algorithm inspired from the explorative behavior of ants. In spite of being blind and weakly intelligent, the ants can find the shortest path from home to the food and vice versa. Biologists found out that this is because of the pheromone trails that they use to communicate and exchange routing data among each other. These trails lead the ants to the shortest possible paths. Ants choose the routes, based on a probability which is proportional to the amount of the pheromones remained on the paths. The stronger the pheromone trail, the fittest the path. This algorithm has some compelling features such as: positive feedback, distributed computation, and a constructive greedy heuristic, which have attracted the researchers [22]. Positive feedback brings about a faster speed to find good solutions. Besides that, distributed computation stops the algorithm from premature and early convergence. And finally, the greedy heuristic helps in finding acceptable solutions in early stages of the search. These are the characteristics which have made the Ant Colony Algorithm robust, versatile and controllable.

3. Proposed approach

The proposed method consists of three main stages including: feature ranking using Fisher criterion, optimum feature subset selection using the hybrid method of cellular learning automata and ant colony, and final feature determination using Receiver Operating Characteristics (ROC) curve. Fig. 1 depicts a view of the proposed methodology.

3.1. Feature ranking using Fisher criterion

In this stage, in order to eliminate the weak features, we utilize a ranking method. With regards to the fact that, in recent studies the focus has been on the Fisher information measure, and this metric has proven its robustness against data scarcity [20], in this work, we used Fisher ratio to rank the features. The Fisher ratio is calculated for features using Eq. (1).

$$FR(j) = \frac{\left(\mu_{j1} - \mu_{j2}\right)^2}{{\sigma_{j1}}^2 - {\sigma_{j2}}^2} \tag{1}$$

Where, μ_{jc} is the sample mean of feature j in class c and $\sigma^2{}_{jc}$ is variance of feature j in c. The N features possessing the highest Fisher value are sent to the next stage.

3.2. Cellular learning automata-ant colony optimization feature selection (CLACOFS)

In this stage we analyze a variety of feature subsets. The N best features in the ranking phase are the input, and a subset with the smallest number of features and high discrimination would be the output. To do this, we consider the problem space as a two dimensional grid of cells. The number of cells is the least power of 2 which is greater than N. The neighborhood is considered to be of Moore type, which implies that each cell will have eight neighbors. Likewise, the cells on the left, right, up and down boundaries are considered to be neighbors.

Each cell can have one of the three states of asleep, awake, and dead. At first, all cells are awake. We consider the environment in the cellular automata to be of type Q. In this case, the feedback of the environment to a cell can have three forms of good, average, and bad. The cell would be rewarded or penalized proportional to the environment's feedback.

We assigned an ant to each cell. We used the Fisher values as heuristic information (initial predictions of feature's performance), and their average as the initial amount of pheromones. In each living cell, the ant uses the probability rule in Eq. (2) to choose features. The number of features each ant is authorized to choose is calculated randomly. The performance of the classifier is determined by the features each ant chooses and is used to update the local pheromone. The environment also analyzes each cell and proportionally rewards or penalizes it based on the performance of the classifier; which changes the cell's energy. Dropping the cell's energy level below the threshold causes it to go asleep and if these conditions remain steadily and sequentially for some iteration the cell dies out.

$$P_i^k(t) = \begin{cases} \dfrac{[\tau_i(t)]^a [\eta_i(t)]^\beta}{\sum_{\upsilon \varepsilon j^k} [t_u(t)]^a [\eta_u(t)]^\beta} \text{if } i \varepsilon \mathrm{j}^k \\ 0 \qquad \text{Otherwise} \end{cases} \tag{2}$$