Graph Based Feature Selection Using Symmetrical Uncertainty in Microarray Dataset

Soodeh Bakhshandeh
Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
soodeh.bakhshandeh@gmail.com

Reza Azmi*
Department of Computer Engineering, Alzahra University, Tehran, Iran
azmi@alzahra.ac.ir

Mohammad Teshnehlab
Department of Control Engineering, K. N. Toosi University, Tehran, Iran
teshnehlab@kntu.ac.ir

Received: 30/Apr/2019             Revised: 25/Sep/2019            Accepted: 04/Nov/2019

Abstract
Microarray data using small samples and thousands of genes provides a difficult challenge for researchers. Utilizing gene selection helps to select the most relevant genes from original dataset with the purpose of dimensionality reduction of microarray data as well as increasing the prediction performance. In this paper, a new gene selection method based on community detection technique and ranking the best genes, is proposed. In order to select the best genes, Symmetric Uncertainty calculates the similarity between two genes, and between gene and its class label. In the first phase, this leads to representation of search space in form of graph. In the second phase, the proposed graph is divided into several clusters, using community detection algorithm. Finally, after ranking the genes, the ones with maximum ranks are selected as the best genes. This approach is a supervised/unsupervised filter-based gene selection method, which not only minimizes the redundancy between genes, but also maximizes the relevance of genes and their class labels. Performance of the proposed method is compared with twelve well-known unsupervised/supervised gene selection approaches over twelve microarray datasets using four classifiers including SVM, DT, NB and k-NN. The results illustrate the advantages of the proposed approach.

Keywords: Gene selection; Microarray data; Filter method; Graph-based clustering; Feature Selection.

1. Introduction
In recent years, a new research path has been opened in Bioinformatics and machine learning field. This field contains monitoring thousands of gene expressions for detecting or classifying the specific type of tumor in DNA microarray datasets. Applying machine learning techniques to microarray data results extracting valuable information from dataset and building a model for classifying data into different categories based on the data classes. In training and testing phase, the researchers deal with small samples consisting of thousands of genes which may lead to “curse of dimensionality” [1]. In high dimensional data, many features are irrelevant and redundant which requires large storage space and consequently, have impact on performance and increasing the cost of learning process [2]. For improving the performance of learning model, the dimensionality reduction methods have been introduced in several papers. These methods reduce the cost and risk of over-fitting, but on the other hand they increase the ability of learning model for classifying the high-dimensionality data in different classes.

For reducing the dimensionality of features (genes), data could be transformed from the original space, with data dimension, into a new space with lower dimension, using feature (gene) extraction techniques. Another technique for reducing the dimensionality of features (genes) is feature (gene) selection. If there are n features (genes), the whole search space size will be 2^n. Therefore, the time complexity of feature selection process is 2^n, which is NP hard problem [3]. In past years, some methods, such as feature selection method, have been introduced to find a near-optimal feature (gene) subset. Feature selection is a preprocessing step that identifies and removes irrelevant and redundant genes from the training data. In classification phase, the gene selection method leads to increasing the influence of training data.

The gene selection methods can be classified into filter, wrapper, embedded, and hybrid categories. The goal of filter, which is independent of any learning algorithm, is to reduce the data dimensionality based on the statistical properties of data. Univariate and multivariate strategies are used for evaluating the relevance of each gene to others in filter method. While univariate methods rank the genes individually and independent of other genes, the multivariate methods consider the correlation between genes. Therefore, because the univariate methods ignore the relation

* Corresponding Author
between genes, they are rather fast but with low accuracy compared to multivariate methods [4]. While Laplacian score [5], term variance [6], mutual information [7], and information gain [8] are popular univariate methods, the well-known multivariate methods are mRMR [9], FCBF [10], RRFs [11], UFSAAC [12], RSM [13].

The wrapper method optimizes a predictor for selection process. The predictor evaluates the quality of the selected genes iteratively. Greedy and stochastic search strategies are two categories of wrapper methods. Sequential forward selection and sequential backward selection are two classical greedy search methods and ant colony optimization (ACO) [14], particle swarm optimization (PSO) [15] are two stochastic search methods. Greedy strategy searches based on single track and stochastic method uses the randomness nature of data. The performance of wrapper method is better comparing to filter method because of using learning model, however the computational cost is higher.

The embedded methods work on training a learning model for classification and using it for building the optimal subset of features. Since the embedded approach uses all the genes for training the classifier, the training phase is so time consuming. Kernel penalized SVM (KPSVM) [16], First Order Inductive Learner (FOIL) [17] and SVM-RFE [18] are three examples of the embedded methods.

The last category of gene selection methods is hybrid methods. This method combines the advantages of filter and wrapper methods for improving the performance of the selected genes in classifying phase. Combining of SVM-RFE and mRMR [19], combining correlation-based feature selection (CFS) and Taguchi-genetic algorithm [20] and combining Fisher score with a GA and PSO [21], are some hybrid methods that are introduced during recent years.

In this paper, a filter-based gene selection method is introduced that uses class label and the correlation between the genes for removing the redundant and irrelevant genes. The proposed method works in three steps; graph representation, gene clustering and selecting the best features from each cluster. In the first step, the gene search space is represented as a graph, corresponding to the similarity between features and also the similarity between feature and its class label. Afterwards, in the second step, a clustering algorithm is applied to the constructed graph to divide the genes into several clusters. Finally, the best genes from each cluster are selected as the final gene subset. The proposed method is compared with other feature/gene selection methods (Mutual correlation (MC), fast clustering-based feature selection algorithm (FAST), Graph Clustering with Node Centrality for unsupervised feature selection (GCNC), Unified-Feature Association Map (U-FAM), Feature Selection method with Joint Maximal Information Entropy between features and class (FS-JMIE), a Correlation based Memetic Algorithm (MA-C), Dense Subgraph Finding with Feature Clustering (DSFFC), Distributed dCor-based FS (D^2CORFS), a ReliefF and ACO-based gene selection (RFACO-GS). A hybrid algorithm for feature subset selection in high-dimensional datasets using FICA and IWSSr algorithm (FICA-IWSSr), Greedy Randomized Adaptive Search Procedure (GRASP), and Support Vector Machine Recursive Feature Elimination (FCSVM-REF)). The comparison is done in terms of classification accuracy, number of selected genes, feature reduction, parameter effects and execution time. The novelties of the proposed method lie in the following aspects:

1. Using Symmetric Uncertainty (SU) for representing the search space as a graph: In this paper, a new method is introduced for calculating the weights of graph based on the similarity between two genes and between each gene and its class label by using SU. Changing the value of the parameters in the first step of the proposed method, leads to supervised or unsupervised gene selection method. If class of dataset samples is available, supervised method is used, otherwise unsupervised method is applied. By using this selection method, the results are considerably improved.

2. Using a new ranking method for determining the important nodes in a graph: In this paper a new ranking method is used for identifying the selection of the influential genes in the third step. After applying the ranking method, a meta-heuristic method is used for selecting the best genes in the third step, resulting minimum redundancy between selected genes in the proposed method.

The rest of the paper is organized as follows: Section 2 contains a brief review on other related works. In Section 3, the materials and concepts, which are used in the proposed method, are explained in more detail. In Section 4, the proposed method is presented. In Section 5, in order to demonstrate the advantages of the proposed method, experimental results on some datasets using different classifiers are presented. Additionally, the reasons backing up the efficiency of the proposed method is described. At the end, in Section 6, the conclusion part is provided.

2. Related Works

As the information of redundant genes presents in other genes, these genes do not help the prediction result. The irrelevant genes do not contribute in increasing the predictive accuracy either. Therefore, the goal of gene selection is removing the redundant and irrelevant genes from gene subset. In past years, some gene/feature selection methods have been introduced considering these goals. Some of them have been successful in removing irrelevant features like [22-24] and some others on eliminating both irrelevant and redundant features ([25, 26]).

As mentioned in section 1, filter gene selection methods are successful in case of dealing with large
number of features. There have been also some graph-based feature selection methods, which work on the relation between different features.

In [27], a fast clustering based feature selection algorithm (FAST) for high dimensional data is proposed, which works in two main steps. In the first step, by using graph-theoretic clustering methods, features are divided into different clusters. In the second step, the relationship between all features and the target class is calculated and based on storage of the relation, some features are selected. In this paper, after eliminating irrelevant features, a minimum spanning tree is constructed, which after partitioning it, the representative features are selected.

In [28], an unsupervised feature selection is presented that works based on graph theory. First, a graph is constructed based on the dissimilarity of features. Then the densest subgraph by maximum average weight is identified. As the reduced subgraph contains the features with less average correlation, these features are the selected features. As in constructing the graph, feature dissimilarity is taken into account, so the feature relevancy does not influence the graph.

In [29], for eliminating both redundant and irrelevant features, the densest subgraph and feature clustering are combined. For removing the redundant features, the densest subgraph like [28] is obtained, and for eliminating the irrelevant features, a specific feature clustering is applied to the feature set.

In [30], ant colony optimization (ACO) with a new fitness function is used for a new unsupervised gene selection method that is called MGSACO. In this method, first a graph represents the search space, using similarity between genes. After that, ACO with new fitness function is used for gene selection.

In [31], a graph is constructed using Pearson correlation coefficient measure for representation the features relationship. Then Louvain community detection algorithm is used for clustering the features. In the last phase, a novel ACO-based search strategy is proposed for selecting a feature subset from the cluster group.

In [32], a new feature selection method called Probabilistic Attribute-Value for Class Distinction (Pavicd), is introduced for removing irrelevant and redundant genes. It works on the space of feature values instead of the features’ space. It only requires an evaluation function, for estimating the prediction of class label by one or more features, and a threshold for analysis of relevance.

In [33], by combining Genetic Algorithm (GA) and Local Search (LS), a correlation based memetic framework is introduced. For tuning the population of GA, symmetrical uncertainty measure is used.

In [34], a joint maximal information entropy between features and their classes is used for feature selection process. For measuring the feature subset, a joint maximal information entropy is defined, and a binary particle swarm optimization searches the feature space for finding the best feature subset.

3. Materials and Methods

The proposed approach is based on graph-theoretic principles, clustering and ranking concepts which have been covered in this section.

3.1 Symmetric Uncertainty (SU)

Entropy is the uncertainty measure in the distribution of variable and is defined as:

\[ H(X) = -\sum_{i} P(x_i) \log_2(P(x_i)) \]  

Where \( X \) and \( P(X) \) are discrete variable and probability mass function of \( X \), respectively and \( H(X) \) indicates entropy of variable \( X \). For observing values of another variable \( Y \), the entropy of \( X \) is defined as:

\[ H(X|Y) = -\sum_{i,j} P(y_j) \sum_{i} P(x_i|y_j) \log_2(P(x_i|y_j)) \]  

Where \( P(X) \) and \( P(X|Y) \) denote the prior probabilities for all value of \( X \) and the posterior probabilities of \( X \) given the value of \( Y \), respectively. Information gain between two variables \( X \) and \( Y \) reflects additional information about \( X \) provided by \( Y \), by which the entropy of \( X \) decreases. Information gain is the decrease of the uncertainty of \( X \) after observing \( Y \), and is defined as:

\[ IG(X|Y) = H(X) - H(X|Y) \]  

Where \( H(X) \) and \( H(X|Y) \) have been defined in equation 1 and 2. If \( IG(X|Y) \) is greater than \( IG(Z|Y) \), the correlation between features \( Y \) and \( Z \) is more than correlation between features \( Y \) and \( X \). Information gain is a symmetrical measure and therefore for two features \( X \) and \( Y \), the order of them does not affect the value of IG. \( IG(X|Y) = IG(Y|X) \). As IG is symmetry, it can be used for measuring the correlation between two features.

For normalizing the value of IG, IG should be divided by feature entropies and is known as Symmetric Uncertainty (SU). In some researches, SU is used to evaluate the goodness of features for classification.

The symmetric uncertainty is defined as follows:

\[ SU(X, Y) = \frac{2 IG(X|Y)}{H(X) + H(Y)} \]  

SU value is in range \([0, 1]\). For two features \( X \) and \( Y \), if the value of SU is 1, it indicates that knowing the value of each feature, the value of other feature is predictable. If the value of SU is 0, it indicates the two variables are independent.

For calculating SU in continues features, the discretization process should be done before [35]. In [27], F-Correlation and T-Relevance are defined as following:

**F-Correlation:** The correlation between any pair of features \( F_i \) and \( F_j \) \( \{(F_i, F_j) \in F \land (i \neq j)\} \) is called the F-Correlation of \( F_i \) and \( F_j \) and denoted by \( SU(F_i, F_j) \).

**T-Relevance:** The relevance between the feature \( F_i \in F \) and the target concept \( C \) is referred to as T-Relevance of \( F_i \) and \( C \) and denoted by \( SU(F_i, C) \).
Following these definitions, the below items are realized:
1. The correlation between irrelevant feature and target concept is very weak.
2. For redundant features, the value of F-Correlation will be near 1.

3.2 Community Detection

In real systems, the graph representation of nodes and vertices is not regular. Using clustering approaches, in the real systems, improves our perception of the relations between patterns. In past decades, some clustering approaches have been introduced for detecting clusters (communities) in complex patterns. $k$-means is one of the classical clustering approaches that is very sensitive to initialized parameters. To compensate for its weakness, the novel clustering approaches are focused on community detection ([36]). Community detection is used for grouping nodes, which are shared common or similar property to one community (also called cluster or module). Therefore, community detection field refers to finding groups of nodes that are more internally connected that externally. Based on [37], using community detection leads to information regarding the network structure, its functionality and its compact representation. Community detection concept is used in some fields such as social networks, recommendation systems, Ad-hoc networks and so on.

Louvain community detection method is one of the popular community detection methods that uses [38]. This method is a heuristic method that works based on modularity function maximization. Modularity has been used to compare the partition quality in different methods, and is an objective function, which its optimization is computationally difficult.

Louvain method is used for finding high modularity partitions in a short period of time. The algorithm steps are intuitive and easy to implement. The computational complexity of the algorithm, for $n$ nodes in the graph, is $O(n \log n)$, therefore for large scale networks with some nodes, the algorithm will be rather quick.

Iterative Louvain method is started with a weighted network as input. In the first step, each node of the network is assigned to a community (community No. has the same value of node No. in the beginning). Then, for each node, the gain modularity is calculated based on removing a node from its community to the other community. In the second phase, a new network, based on modularity, is generated. These two steps are repeated till the significant improvement of the network modularity is obtained. The below equation is used for calculating the gain in modularity by moving one node ($i^{th}$ node) into a community $C$:

$$
\Delta Q_i = \frac{1}{2m} \left[ \frac{\sum_{j=1}^{m+1} w_{ij} - (\sum_{j=1}^{m+1} k_j)}{2m} \right] - \left[ \frac{\sum_{j=1}^{m} w_{ij} - (\sum_{j=1}^{m} k_j)}{2m} \right]
$$

Where $\sum_{in}$ is the summation of all the weights of the links inside the community $C$, where $i$ is moving into, $\sum_{tot}$ is the sum of all the weights of the links to nodes in the community $C$, where $i$ is moving into, $k_i$ is the weighted degree of $i$, $k_{ijn}$ is the sum of the weights of the links between $i$ and other nodes in the community that $i$ is moving into, and $m$ is the sum of all of the edge weights in the network.

3.3 Identifying influential nodes

In complex networks, node importance is a basic measure for characterizing and identifying the structure of them, which is also an open issue ([39]). Centrality is a common measure for ranking the nodes of graphs [40]. Different centrality measures have been proposed such as degree centrality, closeness centrality, betweenness centrality, and eigenvector centrality [41]. Although these centrality measures are used in complex networks, however there are some disadvantages [42].

In [42], a new evaluation method is introduced for determining the important nodes based on Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) approach. In this method, Multiple Attribute Decision Making (MADM) for exploring how to identify important nodes is introduced. TOPSIS method chooses the alternatives that have the shortest distance from the positive ideal solution and the farthest distance from the negative-ideal solution, and it operates in four steps. In the first step, the network is constructed based on the connection of nodes. After that, different centrality values are calculated based on network topology. In the third step, based on Euclidean distance, the separation from the positive ideal alternative $S_p^+$ and the separation from the negative ideal alternative $S^-_n$ of nodes is calculated as follow:

$$
S_p^+ = \sum_{j=1}^{m} (v_i - v_j)^2, \quad i = 1, \ldots, m;
$$

$$
S^-_n = \sum_{j=1}^{m} (v_i^* - v_j)^2, \quad i = 1, \ldots, m;
$$

In the final step, the relative closeness to the ideal alternatives $(C_i)$ is calculated and the alternatives with higher $C_i$ are considered as important alternatives.

$$
C_i = \frac{S_p^+}{S_p^+ + S^-_n}, \quad i = 1, \ldots, m
$$

4. Proposed Method

In this paper, a novel gene selection method is introduced which can remove irrelevant and redundant genes in the selection process. The proposed method consists of three steps. In the following sections, these three steps are described in more detail.

4.1 Graph Representation

In all graph-based feature selection methods, the first step is to represent the search space as an undirected graph. In the constructed graph, each feature represents a
node in graph. The gene search space is mapped to a fully connected undirected weighted graph \( G = (F, E, w_{ij}) \). In this graph, \( F = \{F_1, F_2, \ldots, F_n\} \) denotes a set of original genes (features) and \( E = \{(F_i, F_j); F_i, F_j \in F\} \) is the graph edge and \( w_{ij} \) contains the similarity between two features \( F_i \) and \( F_j \), which are connected by edge \( (F_i, F_j) \).

So far, the methods expressed in different papers have been emphasizing on improving performance of graph-based gene selection algorithm. In this paper, on the other hand, a new measurement based on Symmetrical Uncertainty (SU) is proposed. As it was already explained in 3.1, in SU the similarity value between genes and between genes and their class labels are measured. According to SU, for two features, T-Relevance should be as great as possible, and F-Correlation should be as small as possible. In this case, \( w_{ij} \) is defined as follows:

\[
w_{ij} = \begin{cases} 
\beta \text{SU}(F_i, C) + \gamma \text{SU}(F_j, C) - \alpha \text{SU}(F_i, F_j), & \text{if } i \neq j \\
1, & \text{otherwise}
\end{cases}
\]

Where \( \text{SU}(F_i, C) \) is F-Correlation between two genes \( (F_i, F_j) \) and \( \text{SU}(F_i, F_j) \) and \( \text{SU}(F_j, F_j) \) are T-Relevance values between feature \( F_i \) and \( F_j \) and target class \( C \), respectively.

Based on [43], the measure that is sensitive to rotation is not desirable in many applications. As above formula, if \( \beta \neq \gamma \), therefore the value of \( w_{ij} \) is not equal to \( w_{ji} \). For this reason, \( \gamma \) should be equal to \( \beta \). Therefore, the equation 8 could be changed as follows:

\[
\hat{w}_{ij} = \begin{cases} 
\frac{(1-\alpha)}{2} \left( \text{SU}(F_i, C) + \text{SU}(F_j, C) \right) - \alpha \text{SU}(F_i, F_j), & \text{if } i \neq j \\
0, & \text{otherwise}
\end{cases}
\]

For scaling the edge weight into the range \([0,1]\), softmax scaling is used as follows [6]:

\[
\bar{w}_{ij} = \frac{1}{1 + \exp(-\frac{w_{ij}}{\sigma})}
\]

Where \( \bar{w}_{ij} \) is the edge weight between node \( F_i \) and \( F_j \) and, \( \bar{w} \) and \( \sigma \) are the mean and variance of all edge weights in the graph, respectively. After applying this equation, \( \bar{w}_{ij} \) is the normalized edge weight that represents the similarity between two genes, as well as each gene and its class label on the two sides of the edge.

### 4.2 Gene Clustering

The clustering methods try to group the existing data into different clusters. Therefore, the similarity array is inputted to the proposed algorithm, and the output of clustering method is some clusters that each of them contains the seeds with maximum similarity. Over the past few years, some community detection methods are introduced, which could apply the clustering as well. As mentioned in 3.2, Louvain community detection algorithm, performance wise, is one of the fastest community detection algorithms that has been used in the past. Additionally, this method is reasonably easy to implement. Due to these advantages, the method is used in this paper in the gene clustering step. Therefore, in the second step, after constructing the graph, Louvain community detection algorithm is applied to the graph for grouping the nodes (genes) to different clusters based on the normalized weight of graph edges.

#### 4.3 Selecting the Best Genes

After grouping the genes in different clusters, selecting the best genes in each cluster is a major step by removing irrelevant and redundant genes. Therefore, in this step, the main goal is removing the features that have no influence on the result.

According to subsection 3.3, the technique for order performance by similarity to ideal solution (TOPSIS) approach is introduced for determining the important nodes in a graph. In the proposed method, after constructing the graph in the first step and applying community detection in the second step, for selecting the influential nodes in each cluster, TOPSIS is used.

**TOPSIS** method is very efficient and practical for evaluating the importance of nodes.

For using TOPSIS, a new ranking method is used to rank each gene. After ranking each gene, the best genes are selected and used as final gene subset. For selecting the best genes, two rank parameters are used:

1. Ranking each gene without considering its cluster (Total Rank Score - TRS) and
2. Ranking the genes with considering its cluster (Cluster Rank Score - CRS)

**TRS** is independent of the second phase and **CRS** is dependent to the result of the second phase.

For calculating the first rank value (TRS), TOPSIS runs over all genes, based on the graph that is constructed in the first step. By running **TOPSIS** over all genes, the rank of each gene is compared to the other genes. After that, the total rank score is calculated by dividing the distance between \([0, 1]\) to equal intervals, depending on the number of genes. For example, assuming there are five genes \( F_1, F_2, F_3, F_4 \) and \( F_5 \) in the dataset. If after applying **TOPSIS**, the gene ranks are \( 4, 2, 5, 1 \) and \( 3 \), respectively, then the total rank score \( (TRS) \) of each gene is calculated as Table 1.

For calculating the second rank value (CRS), **TOPSIS** runs over each cluster. In this calculation, the subgraphs, which were constructed after applying community detection in the second phase, are used. Applying **TOPSIS** in each subgraph results the rank of each gene in comparison to the other genes, within each community. Thereafter, for each cluster, the rank of each gene is calculated independent of other clusters and, looking like Table 1:

<table>
<thead>
<tr>
<th>Gene</th>
<th>( F_1 )</th>
<th>( F_2 )</th>
<th>( F_3 )</th>
<th>( F_4 )</th>
<th>( F_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gene rank</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>TRS</td>
<td>0.4</td>
<td>0.8</td>
<td>0.2</td>
<td>1</td>
<td>0.6</td>
</tr>
</tbody>
</table>

*Table 1: Calculating Total Rank Score (TRS)*
Algorithm 1 shows the pseudo-code of calculating CRS and TRS.

For selecting the best genes, two methods are used in this paper. The first method is selecting based on TRS and CRS (TC). The second method is based on TRS, CRS and Simulated Annealing (TCS).

1. **Selecting the best genes based on only TRS and CRS (TC):**

As it was described in 3.2, the features in one cluster have maximum similarity. Therefore, in this step, selecting features from one cluster leads to selecting the genes with maximum redundancy. To reach such a target, a gene with maximum TRS value is selected. In this case, the selected gene would not have lower CRS in compare to other genes from the same cluster. If CRS of the new candidate gene is less than the CRS of other selected genes in its cluster, the new gene, depends on the comparison result of $\Delta T V_{\text{norm}}$ and $\theta$. Term Variance (TV) is to represent valuable information; the larger value this parameter has, the more valuable information the gene contains. TV for gene $F_i$ is defined as follows:

$$TV(F_i) = \frac{1}{|n|} \sum_{j=1}^{|n|} (F_{ij} - \bar{F}_i)$$

Where $|n|$ is the number of samples, $F_{ij}$ shows the value of gene $i$ for sample $j$ and $\bar{F}_i$ is the average value of all samples for gene $F_i$.

$\Delta T V$ is the result of dividing the TV of picking a gene from its cluster by TV of not picking it at all. $\Delta T V$ for gene $i$ in cluster $C$ is calculated as:

$$\Delta T V(F_i) = \frac{\sum_{k} TV(F_k)}{\sum TV(F_i)}$$

Where $|m|$ is the number of selected genes in cluster $C$, $\sum_k TV(F_k)$ and $\sum_i TV(F_i)$ are the summation of TV s of the genes in the cluster, $k \in \{1, \ldots, |m|\}$ and $i \in \{1, \ldots, |n|\}$ with the condition of $k \neq i$.

By normalizing $\Delta T V(F_i)$ in range $[0, 1]$, $\Delta T V_{\text{norm}}(F_i)$ is obtained.

For selecting the gene with lower CRS compared to another CRS in its cluster, $\Delta T V_{\text{norm}}(F_i)$ is calculated for the gene and if $\Delta T V_{\text{norm}}(F_i)$ is greater than $\theta$, $F_i$ will be selected.

Algorithm 2 shows the pseudo-code of using TRS and CRS (TC) to select the final subset.

2. **Selecting the best genes based on TRS, CRS and Simulated Annealing (TCS):**

Simulated Annealing (SA) is a probabilistic non-greedy algorithm that explores search space of a problem by annealing from a high to a low temperature state [44]. In this algorithm, moving to better state is accepted in any beginning, but is getting decreased along with the temperature, and thus the algorithm becomes greedier. If a solid heated past melting point and then cooled down, the solid’s structural properties will vary depending on the rate of cooling. By cooling down the liquid slow enough, large crystals will be formed. On the other hand, if the cooling is done quickly (quenched) the crystals will contain defects and imperfections. The algorithm

### Algorithm 1: CRS-TRS-Calculating Algorithm

**input:**
- $F_i$: $p \times n$ matrix, $p$ patterns of $n$ features;
- $\alpha$: Relevance threshold.

**output:**
- $TRS$: Set of CRS.
- $TRS$: Set of TRS.

1: for $i = 1$ to $n$ do
2: for $j = 1$ to $n$ do
3: $w_{ij} = \frac{1}{2}(SD(F_i,C) + SD(F_j,C)) - nSD(F_i,F_j)$
4: end for
5: end for
6: $\bar{w} \leftarrow \text{Softmax}(\bar{w})$
7: $\{\text{cluster}, \ldots, \text{cluster}\} \leftarrow \text{Louvian}(\bar{w})$
8: for $i = 1$ to $n$ do
9: $\text{TRS} \leftarrow \text{Scale(TOPSIS(F_i))}$
10: end for
11: for $k = 1$ to $\ell$ do
12: for $j = 1$ to $n\ell$ do
13: $\text{CRS}_{kj} \leftarrow \text{Scale(TOPSIS(F_{kj}))}$
14: end for
15: end for

### Algorithm 2: Selecting the best genes using TRS and CRS - TC

**input:**
- $TRS$: Set of CRS;
- $TRS$: Set of TRS.

**output:**
- $S_{opt}$: Set of final selected features.

1: $\text{TRS} \leftarrow \text{Sort-Descending(TRS)}$
2: for $i = 1$ to $n$ do
3: $\text{temp} \leftarrow \text{TRS}_i$
4: $k \leftarrow \text{Check-Cluster(temp)}$
5: if $\text{CRS}(\text{temp}) > \text{CRS(all features in cluster k)}$ then
6: $S_{opt} \leftarrow S_{opt} \cup \text{temp}$
7: else
8: if $\Delta T V_{\text{norm}}(\text{temp}) > \theta$ then
9: $S_{opt} \leftarrow S_{opt} \cup \text{temp}$
10: end if
11: end if
12: end for

terminates when the final temperature reaches, or sufficient number of consecutive moves have been rejected.

In SA, at first an initial solution is randomly selected, and it is assumed to be the optimal solution. Subsequently, the cost of the initial solution is computed using the Cost function. While temperature $T$ does not satisfy the termination condition, a neighboring solution of the current optimal solution is selected, and its cost is calculated. If the cost of the newly selected neighboring solution is less than or equal to current optimal solution, the current optimal solution is replaced with a newly selected neighbor solution. If the cost of the neighboring solution is greater than the current optimal solution, a random value $q$ is chosen in the range of $[0, 1]$. In this case, the replacement of the optimal solution is permitted only if a random value $q$ is less than $e^{-\frac{\text{Cost}(F_B) - \text{Cost}(F_A)}{T}}$. 
After temperature $T$ is reduced based on $g(T)$, the same process is continued until $T$ satisfies the termination condition.

Definition of Cost function is the main step of SA. For this purpose, $\text{Sum CRSS} = \text{Cost function in this paper}$, which is calculated as follows:

\[
SCRS = \sum_{i=1}^{c} ICRS_i
\]

Where

\[
ICRS = \frac{\sum_{j=1}^{m} CRS_j}{m}
\]

Where $c$ and $m$ are the number of cluster and the number of genes in each cluster, respectively and $ICRS$ is summation of CRSSs in each cluster.

The goal of this algorithm is reaching maximum value on $SCRS$ applied to the sorted $TRS$.

For using SA, the inputs are $FB$ and $T_0$. $FB$ is an array of 0 and 1. $fb_i$ is 0 when the gene is not selected and 1 when gene is selected. For calculating $FB$, TRSs are sorted in descending manner, and $fb_i$ represents the index of each variable in $FB$, which is calculated based on the sorted TRSs.

$T$ is the initial temperature. This initial temperature should be large enough to allow sufficient transitions to be accepted. The temperature reduction function is defined as a simple iterative function which is the product of $T$ multiplied by a constant $r (g(T) = r \times T)$. In some references, this value is in range $[0.5, 0.99]$.

Algorithm 3 shows the pseudo-code of using SA, TRS and CRS (TCS) for selecting the final subset.

In the last step, for selecting the final subset, two mentioned methods could be used. When using only TRS and CRS in the selection process, $\theta$ parameter should be set, and when using SA, then it is added to the selection process. The main advantages of using SA are the flexibility and ability to approach global optimality in this step. However, the selection process is slower comparing to when using only CRS and TRS. On the other hand, selecting the best value for $\theta$ in the first method, makes it very sensitive and could affect the results.

Fig. 1 shows the schematic diagram of the proposed method.

5. Experimental Results and Discussion

In this section, the performance of the proposed method is compared to twelve other frequently used gene/feature selection methods upon twelve well-known microarray datasets.

For comparing the proposed method with other methods, twelve different DNA microarray datasets are selected. The selected datasets cover a wide spectrum of cancer types and are already used in some other papers. Datasets include nine binary classification types (Blastom, Colon, Gastric, Central Nervous System (CNS), Leukemia 2-class, Diffuse Large-B-Cell Lymphoma (DLBCL), Prostate Tumor, Ovarian, and Brain) and three multi-class (Lymphoma, SRBCT, and Lung Cancer) that are available at [45-47]. Table 2 displays the brief description of used datasets.

![Fig. 1: Illustrating the proposed method on a schematic diagram](image)

Algorithm 3: Selecting the best genes using TRS, CRS and SA – TCS

input:
- $FB = (fb_1 \ldots fb_n)$ where $fb_i \in [0,1]$;
- $T_0$: The initial temperature;
- $r$: temperature reduction constant.

output:
- $v_b$: Combination of genes.

1: Generate an initial solution, $v_i$
2: $v_b \leftarrow v_i$
3: Calculate the cost of initial solution, $Cost(v_b)$
4: while ($T > T_{min}$) or (iteration $< {t_{max}}$) do
5: Randomly select a neighbor solution, $v_n$, of $v_b$ which have one bit different from $v_b$
6: $\Delta Cost = Cost(v_b) - Cost(v_n)$
7: if ($\Delta Cost \leq 0$) then
8: $v_b \leftarrow v_n$
9: else
10: Generate a random number $q$ uniformly in the range $[0,1]$
11: if ($q < e^{-\frac{\Delta Cost}{rT}}$) then
12: $v_b \leftarrow v_n$
13: end if
14: end if
15: $T \leftarrow g(T)$
16: end while
method with Joint Maximal Information Entropy between features and class (FS-JMIE) [34], A Correlation based Memetic Algorithm (MA-C) [33], Dense Subgraph Finding with Feature Clustering (DSFFC) [29], Distributed dCor-based FS (D²CORFS) [51], a ReliefF and ACO-based gene selection (RFACO-GS) [52]. A hybrid algorithm for feature subset selection in high-dimensional datasets using FICA and IWSS algorithm (FICA-IWSS) [53], Greedy Randomized Adaptive Search Procedure (GRASP) [54], and Support Vector Machine Recursive Feature Elimination (FCSVM-REF) [55].

Since the proposed method is a filter-based gene selection and is independent of using any classifier in the gene selection process, for evaluating the performance, the different types of classifiers are used in experiment phase. Different classifiers that are considered in this paper are Support Vector Machine (SVM) [56], Decision Tree (DT) [57], Naïve Bayes (NB) [6] and k-Nearest Neighbor (kNN) [58].

For obtaining the accurate and stable experiment results, the average classification accuracy rate (%), over ten independent runs, is considered. In each run, for training/testing the classifier, 70%/30% of the dataset is used in train/test phase. In the training phase, the final feature subsets are selected using training sets and the learning model is evaluated in the test phase, using testing sets. All experiment results are obtained on a machine with 2.70 GHz CPU and 4GB of RAM.

5.1 Experimental Results

To demonstrate the performance of the proposed method, in this section, the experimental results have been presented in terms of classification accuracy term, feature reduction, and the parameters’ effect and the execution time. In all results, α is 0.4 when calculating the weights of the graph.

The last step in the proposed algorithm is where selecting the best genes by using two methods; TC and TCS. Where ever TC is used, θ is set to 0.7, and where TCS is applied initial temperature (T₀), temperature reduction constant (τ) and termination condition of SA are 100,000, 0.95 and τ < 0.01, respectively.

The accuracy in the selection process of FCSVM-REF is reported by SVM, since that is the mechanism used in the selection process.

5.1.1 Classification accuracy

To demonstrate the advantages of the proposed methods (TC and TCS), the average classification accuracy of it has been compared with other well-known gene selection methods. Table 3 shows the comparison results between the proposed method with twelve gene selection methods. The results are averaged over ten runs of different gene selection methods, using SVM, DT, NB and kNN classifiers. For each classifier in Table 3, the rank of each algorithm in comparison to other methods is presented in parentheses.

As it is indicated in the results, in average, the proposed method using TC achieves the second rank with 87.33%, 87.29%, and 86.03%, using SVM, DT, and kNN, respectively. In average, the proposed method using TCS, obtains the best rank with 89.08%, 87.5%, and 87.14%, using SVM, DT, and kNN classifiers, respectively.

5.1.2 Feature reduction

In Table 4, feature reduction of the proposed method (TC and TCS) has been compared with other feature selection methods over different datasets. As it is presented, that the proposed method (TC or TCS) feature reductions are 83.81%, 84.75%, 83.19%, 84.78%, 81.90%, 75.65%, 89.25%, 83.36%, 83.74%, 94.97%, 85.23%, 94.18% and 85.27% over Blastom, Colon, Gastric, CNS, Leukemia, DLBCL, Prostate Tumor, Ovarian, Lymphoma, Brain, SRBT, Lung Cancer and in average. The average reduction for the proposed method is the highest one with 83.80% for TC and 85.27% for TCS.

5.1.3 Parameters

In the proposed method, α is a user-specified parameter that is used for calculating the graph weights in equation 10. If α = 0, the equation10 will be changed to:

\[ w_{ij} = \frac{1}{2} \times (SU(F_i, F_j) + SU(F_j, F_i)) \]  (16)

In this situation, only T-Relevance (the correlation between each gene and its target class) affects the graph weights, and so, the method will be will be a supervised graph-based gene selection. While α = 1, the graph weights are calculated as follow:

\[ w_{ij} = -SU(F_i, F_j) \]  (17)

Therefore, only F-Correlation between two genes determines the weight of the edge between two nodes and the effect of its target class will be ignored. This leads to changing the proposed method to an unsupervised graph-based gene selection.

In other situation, the value of \( \frac{1 - \alpha}{2} \) depends on the value of α, and so if the value of α is 0.3, the value of \( \frac{1 - \alpha}{2} \) is 0.35. Figure 2 and Figure 3 show the average classification accuracy of the proposed method using TCS over ten runs with different α values using SVM and DT classifiers. In these figures, for different values of α (which cause change on the \( \frac{1 - \alpha}{2} \) value), the average classification accuracy is calculated and represented. For example, using SVM, and α = 0.4, for Colon, Leukemia, Prostate Tumor, Brain, SRBT, Lung Cancer and average, the classification accuracy is 88.53%, 90.54%, 89.67%, 80.70%, 77.73%, 82.31% and 84.91%, respectively. Using DT and α = 0.4, for Colon, Leukemia, Prostate Tumor, Brain, SRBT, Lung Cancer and average, the classification accuracy is 78.48%, 88.92%, 83.45%, 80.67%, 82.88%, 88.27% and 83.78%, respectively.

In Fig. 2, the maximum classification accuracy, for Colon, Leukemia, Prostate Tumor, Brain, SRBT, Lung Cancer and in average, is 88.53% (α = 0.4), 90.54% (α = 0.4), 89.67% (α = 0.4), 81.61% (α = 0.5), 78.45% (α = 0.3), 84.48% (α = 0.5) and 84.91% (α = 0.4), respectively.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Classifier</th>
<th>Average Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Blasston</strong></td>
<td><strong>SVM</strong></td>
<td>86.03%</td>
</tr>
<tr>
<td></td>
<td><strong>DT</strong></td>
<td>86.34%</td>
</tr>
<tr>
<td></td>
<td><strong>NB</strong></td>
<td>86.34%</td>
</tr>
<tr>
<td></td>
<td><strong>KNN</strong></td>
<td>86.34%</td>
</tr>
<tr>
<td><strong>Colon</strong></td>
<td><strong>SVM</strong></td>
<td>85.46%</td>
</tr>
<tr>
<td></td>
<td><strong>DT</strong></td>
<td>85.87%</td>
</tr>
<tr>
<td></td>
<td><strong>NB</strong></td>
<td>85.87%</td>
</tr>
<tr>
<td></td>
<td><strong>KNN</strong></td>
<td>85.87%</td>
</tr>
<tr>
<td><strong>CNS</strong></td>
<td><strong>SVM</strong></td>
<td>84.85%</td>
</tr>
<tr>
<td></td>
<td><strong>DT</strong></td>
<td>84.85%</td>
</tr>
<tr>
<td></td>
<td><strong>NB</strong></td>
<td>84.85%</td>
</tr>
<tr>
<td></td>
<td><strong>KNN</strong></td>
<td>84.85%</td>
</tr>
<tr>
<td><strong>Leukemia</strong></td>
<td><strong>SVM</strong></td>
<td>83.72%</td>
</tr>
<tr>
<td></td>
<td><strong>DT</strong></td>
<td>83.72%</td>
</tr>
<tr>
<td></td>
<td><strong>NB</strong></td>
<td>83.72%</td>
</tr>
<tr>
<td></td>
<td><strong>KNN</strong></td>
<td>83.72%</td>
</tr>
<tr>
<td><strong>DLBCL</strong></td>
<td><strong>SVM</strong></td>
<td>82.07%</td>
</tr>
<tr>
<td></td>
<td><strong>DT</strong></td>
<td>82.07%</td>
</tr>
<tr>
<td></td>
<td><strong>NB</strong></td>
<td>82.07%</td>
</tr>
<tr>
<td></td>
<td><strong>KNN</strong></td>
<td>82.07%</td>
</tr>
<tr>
<td><strong>Prostate</strong></td>
<td><strong>SVM</strong></td>
<td>81.99%</td>
</tr>
<tr>
<td></td>
<td><strong>DT</strong></td>
<td>81.99%</td>
</tr>
<tr>
<td></td>
<td><strong>NB</strong></td>
<td>81.99%</td>
</tr>
<tr>
<td></td>
<td><strong>KNN</strong></td>
<td>81.99%</td>
</tr>
<tr>
<td><strong>Ovarian</strong></td>
<td><strong>SVM</strong></td>
<td>81.06%</td>
</tr>
<tr>
<td></td>
<td><strong>DT</strong></td>
<td>81.06%</td>
</tr>
<tr>
<td></td>
<td><strong>NB</strong></td>
<td>81.06%</td>
</tr>
<tr>
<td></td>
<td><strong>KNN</strong></td>
<td>81.06%</td>
</tr>
<tr>
<td><strong>Lymphoma</strong></td>
<td><strong>SVM</strong></td>
<td>80.86%</td>
</tr>
<tr>
<td></td>
<td><strong>DT</strong></td>
<td>80.86%</td>
</tr>
<tr>
<td></td>
<td><strong>NB</strong></td>
<td>80.86%</td>
</tr>
<tr>
<td></td>
<td><strong>KNN</strong></td>
<td>80.86%</td>
</tr>
<tr>
<td><strong>Brain</strong></td>
<td><strong>SVM</strong></td>
<td>80.21%</td>
</tr>
<tr>
<td></td>
<td><strong>DT</strong></td>
<td>80.21%</td>
</tr>
<tr>
<td></td>
<td><strong>NB</strong></td>
<td>80.21%</td>
</tr>
<tr>
<td></td>
<td><strong>KNN</strong></td>
<td>80.21%</td>
</tr>
<tr>
<td><strong>SBT</strong></td>
<td><strong>SVM</strong></td>
<td>80.05%</td>
</tr>
<tr>
<td></td>
<td><strong>DT</strong></td>
<td>80.05%</td>
</tr>
<tr>
<td></td>
<td><strong>NB</strong></td>
<td>80.05%</td>
</tr>
<tr>
<td></td>
<td><strong>KNN</strong></td>
<td>80.05%</td>
</tr>
<tr>
<td><strong>Lung Cancer</strong></td>
<td><strong>SVM</strong></td>
<td>79.89%</td>
</tr>
<tr>
<td></td>
<td><strong>DT</strong></td>
<td>79.89%</td>
</tr>
<tr>
<td></td>
<td><strong>NB</strong></td>
<td>79.89%</td>
</tr>
<tr>
<td></td>
<td><strong>KNN</strong></td>
<td>79.89%</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>SVM</strong></td>
<td>80.31%</td>
</tr>
<tr>
<td></td>
<td><strong>DT</strong></td>
<td>80.31%</td>
</tr>
<tr>
<td></td>
<td><strong>NB</strong></td>
<td>80.31%</td>
</tr>
<tr>
<td></td>
<td><strong>KNN</strong></td>
<td>80.31%</td>
</tr>
</tbody>
</table>

* Corresponding Author
Table 4: Comparison of feature reduction of different feature selection methods over different datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Proposed Method</th>
<th>MC</th>
<th>FAS</th>
<th>GC</th>
<th>U- FAM</th>
<th>MA</th>
<th>DSF</th>
<th>FC</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC</td>
<td>ICS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blastem</td>
<td>83.8%</td>
<td>75.4%</td>
<td>63.9%</td>
<td>71.4%</td>
<td>69.3%</td>
<td>72.0%</td>
<td>76.9%</td>
<td>70.4%</td>
</tr>
<tr>
<td>Colon</td>
<td>84.7%</td>
<td>82.4%</td>
<td>71.3%</td>
<td>87.5%</td>
<td>81.2%</td>
<td>59.2%</td>
<td>79.1%</td>
<td>73.3%</td>
</tr>
<tr>
<td>Gastric</td>
<td>83.1%</td>
<td>83.1%</td>
<td>65.4%</td>
<td>80.4%</td>
<td>75.9%</td>
<td>70.4%</td>
<td>74.0%</td>
<td>75.9%</td>
</tr>
<tr>
<td>CNS</td>
<td>81.6%</td>
<td>84.7%</td>
<td>72.0%</td>
<td>73.2%</td>
<td>70.9%</td>
<td>70.8%</td>
<td>72.2%</td>
<td>75.4%</td>
</tr>
<tr>
<td>Leukemia</td>
<td>79.9%</td>
<td>81.9%</td>
<td>63.9%</td>
<td>73.0%</td>
<td>80.3%</td>
<td>76.4%</td>
<td>85.4%</td>
<td>74.3%</td>
</tr>
<tr>
<td>DLBCL</td>
<td>70.2%</td>
<td>75.6%</td>
<td>61.4%</td>
<td>60.8%</td>
<td>66.7%</td>
<td>70.4%</td>
<td>75.2%</td>
<td>68.2%</td>
</tr>
<tr>
<td>Prostate  Tumor</td>
<td>88.1%</td>
<td>89.2%</td>
<td>76.9%</td>
<td>78.9%</td>
<td>81.6%</td>
<td>79.5%</td>
<td>73.5%</td>
<td>67.1%</td>
</tr>
<tr>
<td>Ovarian</td>
<td>81.5%</td>
<td>83.3%</td>
<td>78.4%</td>
<td>80.6%</td>
<td>76.9%</td>
<td>90.1%</td>
<td>79.4%</td>
<td>73.4%</td>
</tr>
<tr>
<td>Lymphoma</td>
<td>80.1%</td>
<td>83.7%</td>
<td>68.9%</td>
<td>78.9%</td>
<td>80.4%</td>
<td>73.8%</td>
<td>71.7%</td>
<td>79.1%</td>
</tr>
<tr>
<td>Brain</td>
<td>92.7%</td>
<td>94.9%</td>
<td>81.9%</td>
<td>74.1%</td>
<td>75.3%</td>
<td>80.8%</td>
<td>80.4%</td>
<td>71.3%</td>
</tr>
<tr>
<td>SRBT</td>
<td>85.1%</td>
<td>85.2%</td>
<td>83.5%</td>
<td>78.3%</td>
<td>87.1%</td>
<td>87.2%</td>
<td>70.4%</td>
<td>69.9%</td>
</tr>
<tr>
<td>Lung Cancer</td>
<td>94.1%</td>
<td>93.4%</td>
<td>54.3%</td>
<td>92.4%</td>
<td>83.9%</td>
<td>86.6%</td>
<td>71.0%</td>
<td>80.2%</td>
</tr>
<tr>
<td>Average</td>
<td>83.8%</td>
<td>85.2%</td>
<td>70.7%</td>
<td>79.7%</td>
<td>78.0%</td>
<td>77.0%</td>
<td>76.2%</td>
<td>73.5%</td>
</tr>
</tbody>
</table>

In Fig. 3, the maximum classification accuracy, for Colon, Leukemia, Prostate Tumor, Brain, SRBT, Lung Cancer and in average is 79.35% ($\alpha = 0.3$), 88.92% ($\alpha = 0.4$), 84.28% ($\alpha = 0.3$), 80.67% ($\alpha = 0.4$), 83.92% ($\alpha = 0.6$), 88.27% ($\alpha = 0.4$) and 83.78% ($\alpha = 0.4$), respectively.

$\theta$ is other parameter that is used in the proposed method for selecting the best genes in the final step (Algorithm 2). Selecting the best value for $\theta$ is very sensitive and could affect the results. Fig. 4 shows the average classification accuracy of the proposed method using TC over ten runs with different $\theta$ values using SVM classifier. Therefore, the maximum classification accuracy, for Colon, Leukemia, Prostate Tumor, Brain, SRBT, Lung Cancer and in average is 83.46% ($\theta = 0.7$), 87.30% ($\theta = 0.7$), 89.94% ($\theta = 0.7$), 81.97% ($\theta = 0.8$), 81.74% ($\theta = 0.9$), 81.06% ($\theta = 0.7$) and 83.59% ($\theta = 0.7$), respectively.

5.1.4 Execution time

To analyze the complexity of the proposed method, the computational complexity should be estimated. The proposed method consists of three steps: graph representation, gene clustering and selecting the best genes from each cluster. Therefore, computational complexity of the proposed method could be estimated as follows:

Step 1: For $n$ genes and $p$ samples, calculating the graph edges for all pairs of genes has a computational cost of $O(n^2p)$.

Step 2: For clustering the genes, Louvain community detection algorithm is used with computational cost of $O(n \log n)$.

Step 3: based on [59], the time complexity of TOPSIS could be calculated as: $O(n^2)$ for the first step, $O(n)$ for the second step, $O(n)$ for the third step and $O(1)$ for the final step. This means the computational cost of TOPSIS is $O(n^2 + n + n + 1) = O(n^2)$.

In the proposed method, the TOPSIS is used for ranking the genes without considering their clusters with $O(n^2)$ computational cost, and considering their clusters.
$O(n^2)$ computational cost, and considering their clusters with $O(mk)$ computational cost, where $m (m \leq n)$ is the number of genes in each cluster and $k$ is the number of clusters ($O(mk) \leq O(n^2)$).

After applying TOPSIS, for selecting the best genes, two different methods are used. For the first method (Algorithm 2), all genes will be traced based on TRS and CRS values. Therefore, the computational cost of this method is $O(n)$.

For the second method (Algorithm 3), because of using the selected temperature reduction function and the stop condition of SA, the computational cost of this method is $O(t_{\text{stop}})$.

Consequently, the total computational complexity of the proposed method is $O(n^2 + n \log n + n^2 + (n \log n)) = O(n^2)$. For comparing the execution time of the proposed method (TC and TCS) with other gene selection methods, the average execution times (in milliseconds) of different methods are shown in Table 5. As for filter based gene selection, where the gene selection process is independent of the classifier, only the execution time of the feature selection process is reported. As described in section 2, for univariate feature selection, the dependency between genes will be ignored and so each gene is evaluated independently. For multivariate gene selection, on the other hand, the dependency between genes will affect the gene selection process. Therefore, for univariate gene selection (i.e., FS, ReliefF), the execution time is greater than multivariate (i.e., mRMR, and GCNC).

### 5.2 Discussion

In the past subsections, the performance of the proposed method was compared to other gene selection methods in different experiments. In this section, the reasons behind the efficiency of the proposed method are described briefly.

Increasing the performance of gene selection in microarray data with high-dimensional genes depends on how well the irrelevant and redundant genes are ignored when dealing with a dataset. If the gene selection method handles both the irrelevant and redundant genes efficiently, then it would obtain better results. That is why when using univariate methods (i.e., LS, FS and ReliefF), since they ignore the gene dependency in their gene selection processes, they are not successful in removing redundant genes. In the proposed method, the symmetric uncertainty concept takes into consideration the relation between genes and also between each gene and its class label. The higher relation between different genes has a negative impact in the gene selection process, but on the other hand the effect of relation between each gene and class label is positive. Therefore, in the proposed method, using SU leads to ignoring the redundant and irrelevant genes, when selecting the genes in the first phase of the process. But using SU for calculating the graph weights in the first step of the proposed method may be time consuming for the datasets with huge number of features.

When it comes to the second phase by using graph clustering algorithm, not only the similar genes within a cluster are grouped, but also genes with minimum dependency to other genes within a cluster are selected. This helps selecting the genes with minimum redundancy, in the proposed method. By ranking the genes in third step and selecting the genes with highest rank on the global and lowest rank inside the cluster, the genes with similar attributes are ignored. Therefore, the redundant and irrelevant genes have a low chance for being picked up to the final feature subsets. In the proposed method, for selecting the best genes in third step, two different methods are used. The first method selects the genes within a cluster based on its probability. In order to calculate the probability, TV has been used. If a gene has greater TV, it means it contains valuable information. Therefore, using $\Delta TV$ (comparing between TV of when the gene is selected and when it is not) as probability of

<table>
<thead>
<tr>
<th>Method</th>
<th>Colon</th>
<th>Leukemia</th>
<th>Prostate</th>
<th>Brain</th>
<th>SRBT</th>
<th>Lung</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC</td>
<td>109,532</td>
<td>189,472</td>
<td>183,944</td>
<td>247,936</td>
<td>140,368</td>
<td>218,629</td>
<td>181,647</td>
</tr>
<tr>
<td>TCS</td>
<td>118,945</td>
<td>210,490</td>
<td>286,701</td>
<td>404,518</td>
<td>172,184</td>
<td>281,309</td>
<td>247,375</td>
</tr>
<tr>
<td>ReliefF</td>
<td>1,978</td>
<td>2,591</td>
<td>2,671</td>
<td>2,734</td>
<td>2,080</td>
<td>2,695</td>
<td>2,458</td>
</tr>
<tr>
<td>FS</td>
<td>89</td>
<td>103</td>
<td>176</td>
<td>217</td>
<td>91</td>
<td>205</td>
<td>147</td>
</tr>
<tr>
<td>LS</td>
<td>1,368</td>
<td>1,498</td>
<td>3,527</td>
<td>8,926</td>
<td>1,395</td>
<td>6,719</td>
<td>3,906</td>
</tr>
<tr>
<td>mRMR</td>
<td>9,763</td>
<td>11,261</td>
<td>13,893</td>
<td>14,789</td>
<td>10,428</td>
<td>14,735</td>
<td>12,478</td>
</tr>
<tr>
<td>RSM</td>
<td>68</td>
<td>71</td>
<td>128</td>
<td>459</td>
<td>73</td>
<td>193</td>
<td>165</td>
</tr>
<tr>
<td>MC</td>
<td>27</td>
<td>64</td>
<td>124</td>
<td>278</td>
<td>59</td>
<td>161</td>
<td>119</td>
</tr>
<tr>
<td>FAST</td>
<td>117,591</td>
<td>162,814</td>
<td>168,436</td>
<td>283,612</td>
<td>154,31</td>
<td>220,581</td>
<td>190,607</td>
</tr>
<tr>
<td>GCNC</td>
<td>116,831</td>
<td>193,477</td>
<td>205,716</td>
<td>259,332</td>
<td>124,821</td>
<td>238,900</td>
<td>189,846</td>
</tr>
<tr>
<td>U-FAM</td>
<td>43,953</td>
<td>61,704</td>
<td>72,136</td>
<td>121,742</td>
<td>51,826</td>
<td>91,491</td>
<td>73,809</td>
</tr>
<tr>
<td>FS-JMIE</td>
<td>186,947</td>
<td>261,955</td>
<td>37,827</td>
<td>450,970</td>
<td>190,333</td>
<td>480,729</td>
<td>268,127</td>
</tr>
<tr>
<td>DSFPC</td>
<td>121,487</td>
<td>218,706</td>
<td>197,467</td>
<td>390,041</td>
<td>120,963</td>
<td>378,117</td>
<td>237,797</td>
</tr>
<tr>
<td>FCSVM-REF</td>
<td>147,281</td>
<td>269,405</td>
<td>293,863</td>
<td>458,052</td>
<td>193,924</td>
<td>394,589</td>
<td>292,852</td>
</tr>
</tbody>
</table>

Table 5: Average execution time (in ms) of different gene selection methods over ten independent runs.
selecting or ignoring a gene, results picking up only those genes which would lead to improve the method’s performance.

In the second method, for selecting the best genes, SA is used, in the proposed method. SA is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a meta-heuristic to approximate global optimization in a large search space. It is often used when the search space is discrete. For problems where finding an approximate global optimum is more important than finding a precise local optimum in a fixed amount of time, SA may be preferable to alternatives. Using Sum CRS (SCRS) as SA cost function leads to decreasing the redundancy between selected genes and increasing the performance of proposed method. The experimental results show that using SA in the third step of the proposed method (TCS) actually improves the results, however the selection process is slower than TC.

The proposed method uses class label in the first step with parameter $\alpha$. This parameter is used for determining the effect of the class label and the relation between different genes. By changing the value of $\alpha$, different conditions will be created and, so, different results will be achieved. For example, if $\alpha = 1$, the class label will be ignored, and the algorithm will be changed to an unsupervised algorithm. This condition would be used when the label of the class is not reliable in the dataset. The Figure 2 and Figure 3 show that if the value of $\alpha$ is in range $[0.3, 0.5]$ ($\frac{1-\alpha}{2}$ in range $[0.25, 0.35]$), the best results are obtained.

5.3 Statistical Analysis

The Friedman test is a non-parametric equivalent that can be used for illustrating the statistical significance of the results over multiple datasets. The test ranks each classifier accuracy over different datasets, separately. Therefore, the best performing algorithm gets the first rank, the second best gets the second rank and so on. After ranking each classifier over different datasets, $\chi_{k}^{2}$ and $F_{p}$ could be calculated. For $N$ datasets and $k$ methods:

$$\chi_{k}^{2} = \frac{12N}{k(k+1)} \times \left[\sum_{i=1}^{k} R_{i}^{2} - \frac{k(k+1)^2}{4}\right]$$ (18)

$$F_{p} = \frac{(N-1)\chi_{k}^{2}}{N(k-1) - \chi_{k}^{2}}$$ (19)

Where $R_{i}^{2}$ is the average rank of the $j^{th}$ method over all datasets and which is distributed according to the Fisher distribution with $k-1$ and $(k-1) \times (N-1)$ degrees of freedom. In Friedman test, using significance level $\alpha$, the null hypothesis means all methods perform equivalently at level $\alpha$.

In our experiments, using SVM, with $N = 12$ and $k = 14$, the critical value of Fisher distribution with 13 and degrees of freedom 143, for $\alpha = 0.05$, $F(13,143)$ is near 1.8. Therefore, if the gene selection method is incorporated with a classifier and the value of $F_{p}$ is greater than 1.836, the null hypothesis will be rejected, and the result is statistically significant.

For other classifiers, with $N = 12$ and $k = 13$, the critical value of Fisher distribution with 12 and degrees of freedom 132, for $\alpha = 0.05$, $F(12,132)$ is near 1.8. Therefore, if the gene selection method is incorporated with a classifier and the value of $F_{p}$ is greater than 1.874, the null hypothesis will be rejected, and the result is statistically significant.

For comparing the results of Friedman test between the proposed method (TC and TCS) and other gene selection methods, after ranking different gene selection method, the results of Friedman test are shown in Table 7. For all classifiers, it is demonstrated that when the gene selection methods are incorporated with different classifiers, the value of $F_{p}$ is greater than 1.8. Therefore, the null hypothesis will be rejected, and it can be concluded that these results are statistically significant.

For demonstrating the distribution of accuracies for different treatments based on the obtained ranks, the distribution of accuracies for twelve datasets are investigated. It conclusion the distribution of all accuracies are near normal. In Fig. 5 and Fig. 6, Quantile-Quantile (Q-Q) plot for five datasets (i.e., Colon, Gastric, Ovarian, SRBT and Lung Cancer) are shown. In this plot the linearity of the points suggests the data are normally distributed [60].

6. Conclusion

In this paper, a new supervised/unsupervised filter gene selection was proposed, which was based on the graph clustering and ranking the genes and it selects the best subset of genes from the microarray data. For selecting the genes with minimum redundancy to other genes and maximum relevance to the class label, some concepts were used; Symmetric Uncertainty for creating graph, community detection for grouping the genes in different clusters, and a new method for ranking the genes. It did not use any learning model in the selection process which leaded to decrease the time complexity. The performance of the proposed method was compared to different gene/feature selection methods including Mutual correlation (MC), fast clustering-based feature selection algorithm (FAST), Graph Clustering with Node Centrality for unsupervised feature selection (GCNC), Unified- Feature Association Map (U-FAM), Feature

<table>
<thead>
<tr>
<th>Classifier</th>
<th>$\chi_{k}^{2}$</th>
<th>$F_{p}$</th>
<th>Significant</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC</td>
<td>TCS</td>
<td>TCS</td>
<td>TCS</td>
</tr>
<tr>
<td>SVM</td>
<td>47.30293</td>
<td>45.08423</td>
<td>8.7425</td>
</tr>
<tr>
<td>DT</td>
<td>47.38560</td>
<td>48.65455</td>
<td>11.7948</td>
</tr>
<tr>
<td>NB</td>
<td>52.84612</td>
<td>50.19547</td>
<td>12.4287</td>
</tr>
<tr>
<td>KNN</td>
<td>48.98744</td>
<td>50.04512</td>
<td>11.7298</td>
</tr>
</tbody>
</table>
Selection method with Joint Maximal Information Entropy between features and class (FS-JMIE), a Correlation based Memetic Algorithm (MA-C), Dense Subgraph Finding with Feature Clustering (DSFFC), Distributed dCor-based FS (D2CORFS), a ReliefF and ACO-based gene selection (RFACO-GS), A hybrid algorithm for feature subset selection in high-dimensional datasets using FICA and IWSSr (algorithm (FICA-IWSSr), Greedy Randomized Adaptive Search Procedure (GRASP), and Support Vector Machine Recursive Feature Elimination (FCSVM-REF). The performance comparison was done over twelve microarray datasets using four different classifiers including Support Vector Machine (SVM), Decision Tree (DT), Naïve Bayes (NB) and k-Nearest Neighbor (kNN).

The experimental results showed that the proposed method using three steps for selecting the best gene subset, was able to select a subset of genes with minimum redundancy between genes and maximum relevance to class label. The results indicated that the proposed method obtained significantly better result in comparison to the other well-known supervised and unsupervised feature/gene selection approaches. The results also confirmed that the proposed method achieved satisfying performance over different classifiers for different microarray datasets.

References


Soodeh Bakhshandeh received his BS degree in Computer Engineering (Software) from Alzahra University, Tehran, Iran in 2007. She graduated in Artificial Intelligence from Science and Research Branch, Islamic Azad University, Tehran, Iran. Her research areas are pattern recognition, feature selection, big data analysis, and data mining.

Reza Azmi received his BS degree in Electrical Engineering from Amirkabir University of technology, Tehran, Iran in 1990 and his MS and PhD degrees in Electrical Engineering from Tarbiat Modares University, Tehran, Iran in 1993 and 1999, respectively. He is a faculty member of Computer Engineering Department of Alzahra University of Technology since 2001. He is conducting research activities in the areas of big data analysis, machine learning and cloud computing.

Mohammad Teshnehlab is a faculty member of Electrical Eng. Department of K. N. Toosi University of Technology. His main research interests focus on Intelligent Systems and Control. Professor Teshnehlab is a member of Industrial Control Center of Excellence and founder of Intelligent Systems Laboratory (ISLab). He also is head and Co-founder of Intelligent Systems Scientific Society of Iran (ISSSI) and member of editorial board of International Journal of Information and Communication Technology Research (IJICTR).