

Cancer Classification using SVM-boosted Multiobjective Differential Fuzzy Clustering

Indrajit Saha¹, Ujjwal Maulik², Sanghamitra Bandyopadhyay³
and Dariusz Plewczynski¹

Abstract. Microarray technology facilitates the monitoring of the expression profile of a large number of genes across different experimental conditions or tissue samples simultaneously. Microarray technology is being utilized in cancer diagnosis through the classification of the tissue samples. In this article, we have presented an integrated unsupervised technique for cancer classification. The proposed method is based on multiobjective differential fuzzy clustering of the tissue samples. In this regard, real coded encoding of the cluster centres is used and two fuzzy cluster validity indices are simultaneously optimized. The resultant set of near-Pareto-optimal solutions contains a number of non-dominated solutions. Each such solution has been improved by a novel technique based on Support Vector Machine (SVM) classification. Thereafter, the final clustering solution is produced by majority voting ensemble technique of all improved solutions. The performance of the proposed multiobjective clustering method has been compared to several other microarray clustering algorithms for three publicly available benchmark cancer data sets, viz., leukemia, Colon cancer and Lymphoma data to establish its superiority. Also statistical significance tests have been conducted to establish the statistical superiority of the proposed clustering method.

1 INTRODUCTION

The advent of microarray technology has made it possible to study the expression profile of a huge number of genes across different experimental conditions or tissue samples simultaneously. This has significant impact on cancer research. Microarray technology is being utilized in cancer diagnosis through the classification of the tissue samples. When microarrays are organized in gene versus sample fashion, then they are very helpful in classification of different types of tissues and identification of those genes whose expression levels are good diagnostic indicators. The microarrays where tissue samples are represent cancerous (malignant) and non-cancerous (benign) cells, then the classification of them will result in binary cancer classification. Most of the researches in the area of cancer diagnosis have focused on supervised classification of cancer data sets through training, validation and testing to classify the tumor samples as malignant or benign, or their subtypes [1, 2]. However, unsupervised classification or clustering of tissue samples

should also be studied since in many cases, labeled tissue samples are not available. In this article, we explore the application of the multiobjective differential fuzzy clustering (MODEFC) for unsupervised classification of cancer data.

A microarray gene expression data, consisting of n genes and T time points, is typically organized in a **2D** matrix $E = [g_{ij}]$ of size $n \times T$. Each element g_{ij} gives the expression level of the i th gene at the j th time point. Clustering [8] is an important microarray analysis tool, is used to identify the sets of genes with similar expression profiles. Clustering methods partition the input space into K regions depending on some similarity/dissimilarity metric where the value of K may or may not be known *a priori*. The main objective of any clustering technique is to produce a $K \times n$ partition matrix $U(X)$ of the given data set X , consisting of n patterns, $X = \{x_1, x_2, \dots, x_n\}$. The partition matrix may be represented as $U = [u_{kj}]$, $k = 1, 2, \dots, K$ and $j = 1, 2, \dots, n$, where u_{kj} is the membership of pattern x_j to the k th cluster. For fuzzy clustering of the data, $0 \leq u_{kj} \leq 1$, i.e., u_{kj} denotes the degree of belongingness of pattern x_j to the k th cluster.

Differential Evolution (DE) [10] has been effectively used to develop single objective clustering techniques [9]. However, a single cluster validity measure is seldom equally applicable for different kinds of data sets. This article poses the problem of fuzzy partitioning as one of multiobjective optimizations (MOO) [6]. Unlike single objective optimization, in MOO, search is performed over a number of, often conflicting, objective functions. The final solution set contains a number of Pareto-optimal solutions, none of which can be further improved on any one objective without degrading it in another. The proposed multiobjective differential fuzzy clustering (MODEFC) algorithm has been optimized the Xie-Beni (XB) index [12] and the fuzzy C-means (FCM) error function J_m [4] simultaneously. Usually, a fuzzy clustering solution is defuzzified by assigning each point to the cluster to which the point has the highest membership degree. In general, it has been observed that for a particular cluster, among the points that are assigned to it based on maximum membership criterion, some have higher membership degree to that cluster, whereas the other points of the same cluster may have lower membership degree. Thus the points in the latter case are not assigned to that cluster with high confidence. This observation motivates us to improve the clustering result obtained by multiobjective fuzzy clustering method using some supervised classification tool, such as Support Vector Machine (SVM) [11], which is trained by the points with high membership degree in a cluster. The trained classifier thereafter can be used to classify the remaining points.

A characteristic of any MOO approach is that it often produces

¹ ICM, University of Warsaw, 02-089 Warsaw, Poland, email: (indra, darman)@icm.edu.pl

² Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, West Bengal, India, email: drumaulik@cse.jdvu.ac.in

³ Machine Intelligence Unit, Indian Statistical Institute, Kolkata-700108, West Bengal, India, email: sanghami@isical.ac.in

a large number of Pareto optimal solutions, from which selecting a particular solution is difficult. The existing methods use the characteristics of the Pareto optimal surface or some external measure for this purpose. However, these approaches almost always pick up one solution from the Pareto optimal set as the final solution, although evidently all the solutions in this set have some information that is inherently good for the problem in hand. Motivated by this observation, this article describes a novel method to obtain the final solution by combining the information of all the non-dominated solutions produced in the final generation. In this approach, first each solution (fuzzy membership matrix) of the final non-dominated front is considered one by one and each of them is improved using SVM classifier [11] as discussed above. Finally, the improved clustering solutions are ensembled to create final optimal clustering solution based on majority voting technique.

The superiority of the proposed MODEFC-SVM algorithm is compared with multiobjective genetic algorithm (non-dominated sorting genetic algorithm-II [6]) based fuzzy clustering algorithm (MOGAFC), integrated version of MOGAFC with SVM and a well-known fuzzy c-means (FCM) [4] algorithm for three publicly available benchmark cancer data sets: Leukemia, Colon cancer, and Lymphoma data. The performance of the proposed clustering scheme is also demonstrated both quantitatively and visually. The superiority of the MODEFC-SVM clustering technique has also been proved to be statistically significant through statistical significance tests.

2 FUNDAMENTALS OF DIFFERENTIAL EVOLUTION AND SVM CLASSIFIER

2.1 Differential Evolution

Differential Evolution [10] is a relatively recent heuristic designed to optimize problems over continuous domains. In DE, each decision variable is represented in the vector by a real number. As in any other evolutionary algorithm, the initial population of DE is generated randomly, and then evaluated. The k th individual vector of the population at time-step (generation) t has d components (dimensions), i.e.,

$$G_k(t) = [G_{k,1}(t), G_{k,2}(t), \dots, G_{k,d}(t)] \quad (1)$$

For each target vector $G_k(t)$ that belongs to the current population, three randomly selected vectors from the current population are used. In other words, the d th component of each trial offspring is generated as follows.

$$\vartheta_k(t+1) = G_i(t) + F(G_n(t) - G_m(t)) \quad (2)$$

Here F is a mutation factor. In order to increase the diversity of the perturbed parameter vectors, crossover is introduced. To this end, the trial vector:

$$Q_k(t+1) = [Q_{k,1}(t+1), Q_{k,2}(t+1), \dots, Q_{k,d}(t+1)] \quad (3)$$

is formed, where

$$Q_{jk}(t+1) = \begin{cases} \vartheta_{jk}(t+1) & \text{if } rand_j(0, 1) \leq CR \text{ or } j = rand(k) \\ G_k(t) & \text{if } rand_j(0, 1) > CR \text{ and } j \neq rand(k) \end{cases} \quad (4)$$

In Eqn. (4), $rand_j(0, 1)$ is the j th evaluation of a uniform random number generator with outcome $\in [0, 1]$. CR is the crossover rate $\in [0, 1]$ which has to be determined by the user. $rand(k)$ is a randomly chosen index $\in \{1, 2, \dots, d\}$ which ensures that $Q_k(t+1)$ gets at least one parameter from $\vartheta_k(t+1)$. The following condition decide whether or not it should become a member of next generation ($t+1$),

$$G_k(t+1) = \begin{cases} Q_k(t+1) & \text{if } f(Q_k(t+1)) < f(G_k) \\ G_k(t) & \text{if } f(Q_k(t+1)) \geq f(G_k) \end{cases} \quad (5)$$

where $f(\cdot)$ is the objective function to be minimized in this article. The processes of mutation, crossover and selection are executed for a fixed number of iterations. The best vector seen up to the last generation provides the solution to the clustering problem. The algorithm of DE is described at here can be used for single objective optimization whereas in multiobjective optimization multiple $f(\cdot)$ are to be optimized simultaneously.

2.2 SVM Classifier

Support Vector Machines (SVM) is a learning algorithm originally developed by Vapnik [11]. It has been used extensively for the purpose of classification in a wide variety of fields. Support Vector Machines classifier is inspired by the statistical learning theory and they perform structural risk minimization on a nested set structure of separating hyperplanes [11]. Fundamentally the SVM classifier is designed for two-class problems. It can be extended to handle multi-class problems by designing a number of one-against-all or one-against-one two-class SVMs.

Suppose a data set consists of n feature vectors $\langle x_i, y_i \rangle$, where $y_i \in \{+1, -1\}$, denotes the class label for the data point x_i . The problem of finding the weight vector w can be formulated as minimizing the following function:

$$L(w) = \frac{1}{2} \|w\|^2 \quad (6)$$

subject to

$$y_i[w \cdot \phi(x_i) + b] \geq 1, \quad i = 1, \dots, n. \quad (7)$$

Here, b is the bias and the function $\phi(x)$ maps the input vector to the feature vector. The SVM classifier for the case on linearly inseparable data is given by,

$$f(x) = \sum_{i=1}^n y_i \beta_i K(x_i, x) + b \quad (8)$$

Where K is the kernel matrix, and n is the number of input patterns having nonzero values of the Lagrangian multipliers (β_i). In case of categorical data, x_i is the i th sample, and y_i is the class label. These n input patterns are called support vectors, and hence the name support vector machines. The Lagrangian multipliers (β_i) can be obtained by maximizing the following:

$$Q(\beta) = \sum_{i=1}^n \beta_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j \beta_i \beta_j K(x_i, x_j), \quad (9)$$

subject to

$$\sum_{i=1}^n y_i \beta_i = 0 \text{ and } 0 \leq \beta_i \leq C, \quad i = 1, \dots, n \quad (10)$$

Only a small fraction of the β_i coefficients are nonzero. The corresponding pairs of x_i entries are known as support vectors and they fully define the decision function. Geometrically, the support vectors are the points lying near the separating hyperplane. $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ is called the *kernel function*. The kernel function may be linear or non-linear, like polynomial, sigmoidal, radial basis functions (RBF), etc. RBF kernels are of the following form:

$$K(x_i, x_j) = e^{-w|x_i - x_j|^2} \quad (11)$$

where x_i denotes the i th data point and w is the weight. In this article, the above mentioned RBF kernel is used. Also, the extended version of the two-class SVM that deals with multi-class classification problem by designing a number of one-against-all two-class SVMs, is used here.

3 MULTIOBJECTIVE OPTIMIZATION AND EXISTING TOOLS

3.1 Mathematical Background

The multiobjective optimization can be formally stated as following: Find the vector $\bar{x}^* = [x_1^*, x_2^*, \dots, x_n^*]^T$ of decision variables which will satisfy the m inequality constraints:

$$g_i(\bar{x}) \geq 0, \quad i = 1, 2, \dots, m, \quad (12)$$

the p equality constraints

$$h_i(\bar{x}) \geq 0, \quad i = 1, 2, \dots, p, \quad (13)$$

and optimizes the vector function

$$\bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), \dots, f_k(\bar{x})]^T \quad (14)$$

The constraints given in Eqns. 12 and 13 define the feasible region $\mathcal{F}$ which contains all the admissible solutions. Any solution outside this region is inadmissible since it violates one or more constraints. The vector $\bar{x}^*$ denotes an optimal solution in $\mathcal{F}$. In the context of multiobjective optimization, the difficulty lies in the definition of optimality, since it is only rare that we will find a situation where a single vector $\bar{x}^*$ represents the optimum solution to all the objective functions.

The concept of *Pareto optimality* comes handy in the domain of multiobjective optimization. A formal definition of Pareto optimality from the viewpoint of minimization problem may be given as follows: A decision vector $\bar{x}^*$ is called Pareto optimal if and only if there is no $\bar{x}$ that dominates $\bar{x}^*$, i.e., there is no $\bar{x}$ such that

$$\forall i \in \{1, 2, \dots, k\}, f_i(\bar{x}) \leq f_i(\bar{x}^*) \quad (15)$$

and

$$\exists i \in \{1, 2, \dots, k\}, f_i(\bar{x}) < f_i(\bar{x}^*) \quad (16)$$

In other words, $\bar{x}^*$ is Pareto optimal if there exists no feasible vector $\bar{x}$ which causes a reduction on some criterion without a simultaneous increase in at least another. In this context, two other notions viz., *weakly non-dominated* and *strongly non-dominated* solutions are defined [5]. A point $\bar{x}^*$ is a weakly non-dominated solution if there exists no $\bar{x}$ such that $f_i(\bar{x}) < f_i(\bar{x}^*)$, for $i = 1, 2, \dots, k$. A point $\bar{x}^*$ is a strongly non-dominated solution if there exists no $\bar{x}$ such that $f_i(\bar{x}) \leq f_i(\bar{x}^*)$, for $i = 1, 2, \dots, k$, and for at least one i , $f_i(\bar{x}) < f_i(\bar{x}^*)$. In general, Pareto optimum usually admits a set of solutions called non-dominated solutions.

3.2 Non-dominated Sorting

In this approach, every solution from the population is checked with a partially filled population of dominance. To start with, the first solution from the population is kept in the set S . Thereafter, each solution p (the second solution onwards) is compared with all members of the set S one by one. If the solution p dominates any member q of S , then solution q is removed from S . These way non-members of the non-dominated front get deleted from S . Otherwise, if solution p is dominated by any member of S , then solution p is ignored. If solution p is not dominated any member of S , it is entered in S . This is how the set S grows with non-dominated solutions. When all solutions of the population are checked, the remaining members of S constitute the non-dominated set. To find the other fronts, the members of S will be discounted and the above procedure is repeated. After getting the different fronts, they are assigned through different ranks. The maximum computational complexity of this approach to find the non-dominated front is $O(\log(M \times N^2))$ where M is the number of objective and N is the size of the population [6].

3.3 Crowding Distance

For estimating the density of the solutions surrounding a particular solution in the population, the average distance of two points on either sides of this point along each of the objectives are computed. This quantity $j_{distance}$ serves as an estimate of the size of the largest cuboid enclosing the point j without including any other point in the population, is called *crowding distance*. In Fig. 1, the crowding distance of the j th in its front is the average side-length of the cuboid. The crowding distance computation requires sorting of the population according to each objective function value in their ascending order of magnitude. Thereafter, for each objective function, the boundary solutions are assigned as infinity distance value. All other intermediate solutions are assigned a distance value equal to the absolute difference in the functional values of two adjacent solutions. This computation is continued with other objective functions. The overall crowding distance value is calculated as the sum of individual distance values corresponding to each objective. Computational complexity of crowding distance is $O(M \times N \times \log(N))$ [6].

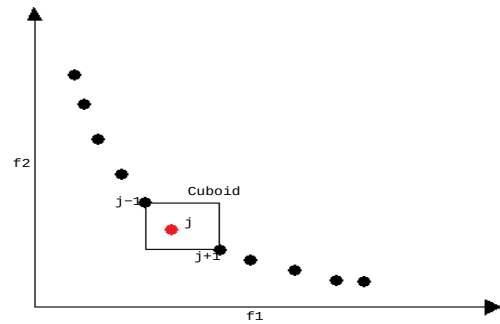


Figure 1. The crowding distance calculation is shown.

4 PROPOSED MULTIOBJECTIVE ALGORITHM

4.1 Multiobjective Differential Evolution based Fuzzy Clustering

4.1.1 Vector Representation and Initial Population

In Multiobjective Differential Evolution based Fuzzy Clustering (MODEFC), each vector is a sequence of real numbers representing the K cluster centres. For an d -dimensional space, the length of a vector is $l = d \times K$, where the first d positions represent the first cluster centre, the next d positions represent those of the second cluster centre, and so on. The K cluster centres encoded in each vector are initialized to K randomly chosen points from the data set. This process is repeated for each of the P vectors in the population, where P is the size of the population.

4.1.2 Evaluation of the Objectives

For computing the objective functions, first the centres $V = \{v_1, v_2, \dots, v_K\}$ encoded in a given chromosome are extracted. The fuzzy membership values u_{ki} , $k = 1, 2, \dots, K$, $i = 1, 2, \dots, n$ are computed using the following equation:

$$u_{ki} = \frac{\left(\frac{1}{D(v_k, x_i)}\right)^{\frac{1}{m-1}}}{\sum_{j=1}^K \left(\frac{1}{D(v_j, x_i)}\right)^{\frac{1}{m-1}}}, \text{ for } 1 \leq k \leq K, 1 \leq i \leq n \quad (17)$$

where $D(v_k, x_i)$ and $D(v_j, x_i)$ are the distances between x_i and v_k , and x_i and v_j respectively. The value of m , the fuzzy exponent, is taken as 2. Note that while computing u_{ki} using Eqn. 17, if $D(v_j, x_i)$ is equal to zero for some j , then u_{ki} is set to zero for all $k = 1, \dots, K$, $k \neq j$, while u_{ki} is set equal to one. Based on the membership values, the cluster centres are recomputed using the following equation

$$v_k = \frac{\sum_{i=1}^n u_{ki}^m x_i}{\sum_{i=1}^n u_{ki}^m} \quad 1 \leq k \leq K \quad (18)$$

The membership values are then recomputed using Eqn. 17. The XB index is defined as a function of the ratio of the total variation σ ($\sigma(U, Z; X) = \sum_{k=1}^K \sum_{i=1}^n u_{ki}^2 D^2(v_k, x_i)$) to the minimum separation sep ($sep(Z) = \min_{k \neq j} D^2(v_k - v_j)$) of the clusters. The XB index is then written as:

$$XB(U, Z; X) = \frac{\sigma(U, Z; X)}{n \times sep(Z)} \quad (19)$$

Note that when the partitioning is compact and the clusters are well separated, value of σ should be low while sep should be high, thereby yielding lower values of the XB index. The objective is therefore to minimize it.

The other objective is the J_m validity measure that is optimized by the FCM algorithm. This computes the global fuzzy variance of the clusters and this is expressed by the following equation:

$$J_m = \sum_{i=1}^n \sum_{k=1}^K u_{ki}^m D^2(v_k, x_i) \quad (20)$$

J_m is to be minimized to get compact clusters.

XB and J_m indices are used as they are contradictory in nature. XB index is responsible for both compactness and separation for the clusters, whereas J_m only represents the global compactness of the clusters.

4.1.3 Other processes

After evaluating the fitness of each vector, it goes through mutation (described in Eqn. 2) to generate the new offspring and crossover (described in Eqn. 4) for increasing the diversity of the mutant vector. The created offspring pool combined with its parent pool for performing the non-dominated sort in the next step. Thereafter, the selection process has been performed based on the lowest rank assigned by the non-dominated sort as well as least crowding distance. These processes are executed for a fixed number of iterations and final non-dominated front is used in the next section for further processing.

4.2 Integrating MODEFC with SVM and Voting Ensemble

Step1: Apply MODEFC on the given data set to obtain a set $S' = \{s_1, s_2, \dots, s_N\}$, $N \leq P$, (P is the population size) of non-dominated Pareto-optimal solution strings consisting of cluster centres.

Step2: Substeps of Step 2 are repeated N times depending on the size of non-dominated solutions s_i , $1 \leq i \leq N$.

- Select 50% of data points from each cluster which are nearest to the respective cluster centres for each s_i .
- Train the SVM classifier by the selected training points.
- Predict the class labels for the remaining points (test points) using the trained SVM classifier.
- Combine the label vectors corresponding to training and testing points to obtain the label vector δ_i for the complete data set corresponding to the non-dominated solution s_i .

Step3: Reorganize the data points to make them consistent with each other, i.e., cluster j in the first solution should be equivalent to cluster j in all the other solutions. For example, the solution string $\{(p, q, r), (a, b, c)\}$ is equivalent to $\{(a, b, c), (p, q, r)\}$. The reorganization is done in such way that each δ_i , $i \neq 1$, becomes consistent with δ_1 .

Step4: Apply majority voting on the label vectors δ_i , $i = 1, \dots, N$ to obtain the final clustering label vector δ . The majority voting is done as follows: assign each point $k = 1, \dots, n$ to the cluster j where the label j appears the maximum number of times among all the labels for the point k in all the δ_i .

5 DATA SETS AND PREPROCESSING

In this article, three publicly available benchmark cancer data sets, viz., Leukemia, Colon cancer and Lymphoma data sets have been used for experiments. The data sets and their pre-processing are described in this section.

5.1 Leukemia Data

The Leukemia data set [7] consists of 72 tissue samples. The samples consist of two types of leukemia, 25 of AML and 47 of ALL. The samples are taken from 63 bone marrow samples and 9 peripheral blood samples. There are 7,129 genes in the data set. The data set is publicly available at <http://www.genome.wi.mit.edu/MPR>.

The data set is subjected to a number of pre-processing steps to find out the genes with most variability. As this article considers the problem of unsupervised classification, hence the gene selection steps followed here are also completely unsupervised. However,

more sophisticated methods for gene selection could have been applied. First we select the genes whose expression levels fall between 100 and 15000. From the resulting 1015 genes, the 100 genes with the largest variation across samples are selected, and the remaining expression values are log-transformed. The resultant data set is of dimension 72×100 .

5.2 Colon Cancer Data

The Colon cancer data set [2] consists of 62 samples of colon epithelial cells from colon cancer patients. The samples consists of tumor biopsies collected from tumors (40 samples), and normal biopsies collected from healthy part of the colons (22 samples) of the same patient. The number of genes in the data set is 2000. The data set is publicly available at <http://microarray.princeton.edu/oncology>.

This data set is pre-processed as follows: first the genes whose expression levels fall between 10 and 15000 are selected. From the resulting 1756 genes, the 100 genes with the largest variation across samples are selected, and the remaining expression values are log-transformed. The resultant data set is of dimension 62×100 .

5.3 Lymphoma Data

The diffuse large B-cell lymphoma (DLBCL) dataset [1] contains expression measurements of 96 normal and malignant lymphocyte samples each measured using a specialized cDNA microarray, containing 4,026 genes that are preferentially expressed in lymphoid cells or which are of known immunological or oncological importance. There are 42 DLBCL and 54 other cancer disease samples. The data set is publicly available at <http://genome-www.stanford.edu/lymphoma>.

The pre-processing steps for this data sets are as follows: As the data set contains some missing values, we select only those genes which do not contain any missing value. This results in 854 genes. Next each gene is normalized to have expression value between 0 and 1. Thereafter top 100 genes with respect to variance are selected. Hence the data set contains 96 samples each described by 100 genes.

6 EXPERIMENTAL RESULTS

6.1 Performance Metric

Silhouette index is a cluster validity index that is used to judge the quality of any clustering solution C . Suppose a represents the average distance of a gene from the other genes of the cluster to which the gene is assigned, and b represents the minimum of the average distances of the gene from the genes of the other clusters. Now the silhouette width s of the gene is defined as:

$$s = \frac{b - a}{\max\{a, b\}} \quad (21)$$

Silhouette index $s(C)$ is the average Silhouette width of all the genes and it reflects the compactness and separation of clusters. The value of Silhouette index varies from -1 to 1 and higher value indicates better clustering result.

6.2 Distance Measure

Pearson Correlation: Given two sample vectors, α_i and α_j , Pearson correlation coefficient $Cor(\alpha_i, \alpha_j)$ between them is computed as:

$$Cor(\alpha_i, \alpha_j) = \frac{\sum_{l=1}^p (\alpha_{il} - \mu_{\alpha_i})(\alpha_{jl} - \mu_{\alpha_j})}{\sqrt{\sum_{l=1}^p (\alpha_{il} - \mu_{\alpha_i})^2} \sqrt{\sum_{l=1}^p (\alpha_{jl} - \mu_{\alpha_j})^2}}. \quad (22)$$

Here μ_{α_i} and μ_{α_j} represent the arithmetic means of the components of the sample vectors α_i and α_j respectively. Pearson correlation coefficient defined in Eqn. 22 is a measure of similarity between two samples in the feature space. The distance between two samples α_i and α_j is computed as $1 - Cor(\alpha_i, \alpha_j)$, which represents the dissimilarity between those two samples.

6.3 Input Parameters

The population size and number of generation used for MODEFC and MOGAFC are 50 and 100 respectively. The crossover probability and mutation factors (F) for MODEFC are set to be 0.8 and 0.7, respectively. The crossover and mutation probabilities for MOGAFC are taken to be 0.8 and 0.3, respectively. The FCM algorithm is executed till it converges to the final solution. For all the fuzzy clustering algorithms m , the fuzzy exponent, is set to 2.0. The results provided corresponding to the MODEFC and MOGAFC are obtained by selecting the solution of the final non-dominated front that provides the best Silhouette index. Results reported in the tables are the average values obtained over 50 runs of the algorithms.

Table 1. Average $s(C)$ values over 50 runs for the three microarray data sets.

Algorithms	Leukemia	Colon cancer	Lymphoma
MODEFC-SVM	0.3942	0.5173	0.4083
MODEFC	0.3601	0.4284	0.3682
MOGAFC-SVM	0.3129	0.4357	0.3253
MOGAFC	0.2851	0.4006	0.2902
FCM	0.2453	0.1649	0.2239

6.4 Performance

Table 1 reports the average $s(C)$ index scores over these 50 runs, for the Leukemia, Colon cancer and Lymphoma data sets. As is evident from the table, the MODEFC-SVM clustering produces the best $s(C)$ index scores compared to the other algorithms. For example, for the leukemia data set, MODEFC-SVM produces average $s(C)$ index score of 0.3942, whereas the next best $s(C)$ score is 0.3601, provided by MODEFC. Similar results are obtained for all the data sets. In general, it can be noted that MODEFC-SVM consistently outperforms the other algorithms for each of the Leukemia, Colon cancer and Lymphoma data sets in terms of $s(C)$. From the table, it appears that MODEFC-SVM also outperforms single objective counterpart FCM. FCM minimizes the J_m measure. On the other hand MODEFC-SVM minimizes J_m and XB objectives simultaneously. As MODEFC-SVM performs better in terms of cluster validity index ($s(C)$), it establishes that optimizing multiple criteria simultaneously can yield better clustering rather than the cases when the objectives are optimized separately.

For the purpose of illustration, Fig. 2 shows the boxplot representing the $s(C)$ index values, over 50 runs of the algorithms for the three data sets considered here. It is evident from the figure that the boxplot

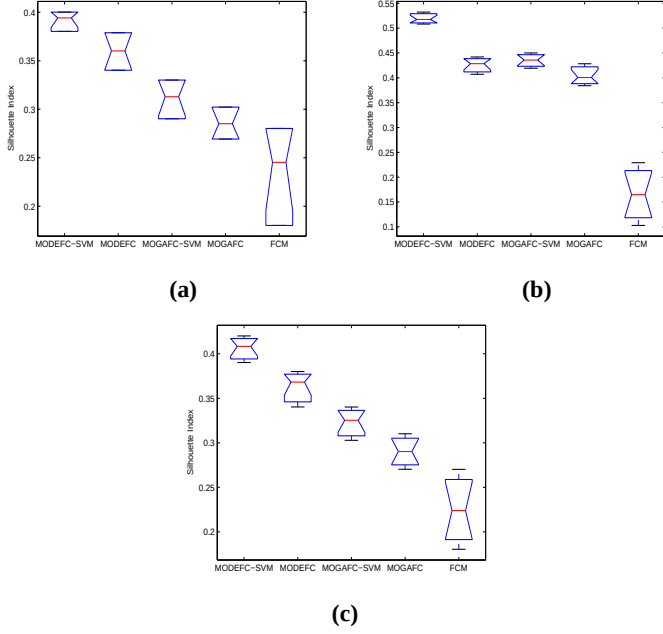


Figure 2. Boxplot of $s(C)$ produces by different clustering algorithms over 50 runs for (a) Leukemia (b) Colon cancer (c) Lymphoma data sets.

corresponding to the MODEFC-SVM clustering method is situated to the upper side of the figure, which indicates that MODEFC-SVM produces higher $s(C)$ index values compared to the other algorithms in all the runs.

7 STATISTICAL SIGNIFICANCE TEST

To establish that MODEFC-SVM is significantly superior compared to the other algorithms, a statistical significance test called t -test has been conducted at the 5% significance level. Six groups, corresponding to the seven algorithms (1. MODEFC-SVM, 2. MODEFC, 3. MOGAFC-SVM, 4. MOGAFC, and 5. FCM) have been created for each data set.

Each group consists of the $s(C)$ index scores produced by 50 consecutive runs of the corresponding algorithm. As evident from the Table 2, the average values of $s(C)$ values for MODEFC-SVM are better than those for the other algorithms. To establish that this goodness is statistically significant, Table 2 reports the P -values produced by t -test for comparison of two groups (group corresponding to MODEFC-SVM and a group corresponding to some other algorithm) at a time. As a null hypothesis, it is assumed that there are no significant differences between the mean values of two groups. Whereas, the alternative hypothesis is that there is significant difference in the mean values of the two groups. All the P -values reported in the table are less than 0.05 (5% significance level). This is strong evidence against the null hypothesis, indicating that the better mean values of the $s(C)$ index produced by MODEFC-SVM is statistically significant and has not occurred by chance.

8 CONCLUSION

In this article, an unsupervised cancer data classification technique based on multiobjective differential fuzzy clustering has been

Table 2. P -values produced by t -test comparing MODEFC-SVM with other algorithms.

Data Set	P -value			
	MODEFC	MOGAFC-SVM	MOGAFC	FCM
Leukemia	0.0313	1.5151e-003	1.4147e-004	5.1307e-005
Colon cancer	0.0406	1.6336e-003	1.5303e-004	5.9084e-005
Lymphoma	0.0382	1.5372e-003	1.4815e-004	4.8203e-005

developed. Real coded encoding of the cluster centres has been used and two cluster validity measures, namely XB validity index and FCM measure J_m have been simultaneously optimized. The non-dominated solutions produced at the final generation are improved through SVM classification. A novel technique for producing the final clustering result by combining the clustering information of the improved solutions through a majority voting ensemble method is proposed. Results on three publicly available benchmark cancer data sets, viz., Leukemia, Colon cancer and Lymphoma, have been demonstrated. The performance of the proposed technique has been compared with that of MODEFC, MOGAFC-SVM, MOGAFC and FCM clustering methods. The results have been demonstrated both quantitatively and visually. The developed MODEFC-SVM clustering technique consistently outperformed the other algorithms considered here. Also statistical superiority has been established through statistical significance test.

ACKNOWLEDGEMENTS

This work was supported by the Polish Ministry of Education and Science (grants: N301 159735, N518 409238).

REFERENCES

- [1] A. A. Alizadeh, M. B. Eisen, R. Davis, C. Ma, I. Lossos, A. Rosenwald, J. Boldrick, R. Warnke, R. Levy, W. Wilson, M. Grever, J. Byrd, D. Botstein, P. O. Brown, and L. M. Straudt, 'Distinct types of diffuse large b-cell lymphomas identified by gene expression profiling', *Nature*, **403**, 503–511, (2000).
- [2] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, 'Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays', in *Proceedings of National Academy of Science, Cell Biology*, **96**, 6745–6750, (1999).
- [3] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum, New York, 1981.
- [4] C. A. Coello Coello, 'A comprehensive survey of evolutionary-based multiobjective optimization techniques', *Knowledge and Information Systems*, **1**(3), 129–156, (1999).
- [5] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan, 'A fast and elitist multiobjective genetic algorithm: NSGA-II', *IEEE Transactions on Evolutionary Computation*, **6**(6), 182–197, (2002).
- [6] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gassenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, D. D. Bloomeld, and E. S. Lander, 'Molecular classification of cancer: class discovery and class prediction by gene expression monitoring', *Science*, **286**, 531–537, (1999).
- [7] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
- [8] U. Maulik and I. Saha, 'Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery', *Pattern Recognition*, **42**(9), 2135–2149, (2009).
- [9] R. Storn and K. Price, 'Differential evolution - A simple and efficient heuristic strategy for global optimization over continuous spaces', *Journal of Global Optimization*, **11**, 341–359, (1997).
- [10] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, USA, 1998.
- [11] X. L. Xie and G. Beni, 'A validity measure for fuzzy clustering', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, 841–847, (1991).