2012 11th IEEE International Conference on Machine Learning and Applications (ICMLA), Pages 33-39

Introducing Subspace Grids to Recognise Patterns in Multidimensinal Data

M. Arif Wani
Computer and Electrical Engineering and Computer Science Department
California State University, USA e-mail: awani@csub.edu

***Abstract*—The work presented in this paper proposes a new approach of using subspace grids for recognizing patterns in multidimensional data. The proposed approach addresses the two problems often associated with this task: i) curse of dimensionality ii) cases with small sample sizes. To handle the curse of dimensionality problem, this paper introduces subspace grids and shows how it can be employed for pattern recognition tasks efficiently. To address the cases with small sample sizes, this paper proposes a multi-scale approach where coarse scale, being stable and generic in nature, suits well for small sample sizes, and fine scales, being more specialized in nature, enhance classification accuracy. The paper first describes projection of multidimensional data to a number of lower dimensional subspaces. Principal component analysis (PCA) and multiple discriminant analysis (MDA) algorithms are used to define lower dimensional subspaces. The range of value associated with each vector of a subspace is divided into a number of equal parts to define coarse subspace grids. Coarse subspace grids are further divided equally into fine subspace grids. A recursive procedure is then employed to obtain rules where coarse and fine subspace grids form premises of rules. The system is tested on the bench mark IRIS data set having 150 examples. (50 examples belonging to each class type). The results show that the use of subspaces grids produces good results to recognize patterns in multidimensional data.**

Keywords- *subspace grids; machine learning; pattern recognition; principle component analysis; multiple discriminant analysis; rule extraction; multi-scale approach..*

I. INTRODUCTION

The work described in this paper is about the task of recognizing patterns in multidimensional data. Two problems often associated with this task are: i) curse of dimensionality ii) cases with small sample sizes.

Curse of dimensionality is normally addressed by projecting the multidimensional data to a lower dimensional space. Thus one of the major tasks in analyzing patterns in multidimensional data, associated with a given application, is to project the data to a lower dimensional space first and then analyzing the data in lower dimensional space. To avoiding the curse of dimensionality problem, this paper introduces subspace grids in lower dimensional space and shows how it can be employed for pattern recognition tasks.

To address the cases with small sample sizes, this work proposes a multi-scale approach where coarse scale is more generic in nature, which suits small sample sizes, and fine scales are more specialized in nature, which further enhance accuracy.

II. LITERATURE REVIEW

A lot of literature is available on classification of multidimensional data. Some of the work is summarized in this section.

A comparative study of pattern selection methods for classification of multidimensional data is presented by Chai and Domeniconi [1]. The authors compare several feature ranking techniques, including variants of correlation coefficients, and Support Vector Machine (SVM) method based on Recursive Feature Elimination (RFE).

A study by Hori *et al*. [2] shows that an independent component analysis (ICA) based method can effectively and blindly classify a vast amount of multidimensional data. Based on the results, authors suggest that the ICA based method can be a powerful approach for classification tasks. The authors also examine classification by principal component analysis (PCA), and compare results of PCA and ICA methods. PCA only takes into account the second-order statistics and restricts itself to orthogonal transformation to obtain principal components. On the other hand, ICA can take into account higher order statistics and can utilize non orthogonal transformation for de-mixing.

Pique-Regil *et al*. [3] propose a sequential Diagonal Linear Discriminant Analysis (SeqDLDA) technique that combines gene selection and classification. At each iteration, one gene is sequentially added and the linear discriminate (LD) recomputed using the SeqDLDA model. Classical Diagonal Linear Discriminant Analysis (DLDA) will add the gene with highest t-test score without checking the resulting model. In contrast, SeqDLDA will find the one

gene that better improves class separation after recomputing the model parameters using a robust t-test score.

A data-dependent kernel for microarray data classification was presented by Xiong *et al*. [4]. This kernel function is engineered so that the class reparability of the training data is maximized. A bootstrapping-based resampling scheme is introduced to reduce the possible training bias.

Wang *et al*. [5] use a hybrid huberized support vector machine (HHSVM). The HHSVM uses the huberized hinge loss function to measure misclassification and the elastic-net penalty to control the complexity of the model. They develop an efficient algorithm that computes the entire regularized solution path for HHSVM.

Kim and Cho [6] proposed two different correlation methods for the generation of feature sets to learn ensemble classifiers. Each ensemble classifier combines several other classifiers that learn from different features. They adopted several feature selection methods, which includes the Pearson's and Spearman's correlation coefficients, the Euclidean distance, the cosine coefficient, information gain, mutual information and signal-to-noise ratio. Experimental results show that two ensemble classifiers whose components are learnt from different feature sets that are negatively or complementarily correlated with each other produce good recognition accuracy rates on the chosen datasets.

A tensor based method to solve the supervised dimensionality reduction problem is presented in [7]. The work first utilizes a multilinear principal component analysis (MPCA) to reduce the tensor object dimension and it then applies a multilinear discriminant analysis (MDA) to find the best subspaces. The number of possible subspace dimensions for any kind of tensor objects is extremely high, so testing all of them for finding the best one is not feasible. The authors address this issue by presenting a method similar to sequential mode truncation (SMT) and full projection is used to initialize the iterative solution to find the best dimension for MDA. The method described by authors saves additional efforts that would be required otherwise to find the best dimension manually.

An incremental approach for microarray classification problem was proposed in [8]. The approach is based on a hybrid principal component analysis (PCA) and multiple discriminant analysis (MDA). The work uses several subspaces, where data is incrementally projected. The resulting incremental hybrid PCA and MDA approach helped in enhancing the classification accuracy of the microarrays.

A subspace grid based approach for recognizing patterns in microarray data was proposed in [9]. The paper first defines a subspace with the aid of principal component analysis (PCA) and multiple discriminant analysis (MDA) algorithms. Each axes of the subspace is divided into equal number of parts to obtain subspace grids. A recursive procedure is then used to obtain rules where subspace grids form premises of rules. The extracted set of rules is evaluated on both training and testing data sets where good results are reported.

This work presents a new approach that incorporates coarse and fine subspace grids for recognizing patterns in multidimensional data. Section III describes the approach used for recognizing patterns using subspace grids. Results and discussion is presented in section IV. Conclusion is finally summarized in section V.

III. SUBSPACE GRIDS FOR PATTERN RECOGNITION

Recognition of patterns in applications that involve multidimensional data is computationally intensive and poses challenges in finding methods that address this issue. Two problems often associated with this task are: i) curse of dimensionality ii) cases with small sample sizes. We propose a multi-scale subspace grid based strategy to address these issues.

To avoiding the 'curse of dimensionality' problem, we introduce a subspace grid based strategy that involves projecting multidimensional data to lower dimensional subspaces and then creating grids to facilitate pattern recognition task in an efficient manner.

To address the 'cases with small sample sizes' problem, the above strategy is refined so that it involves creating multi-scale grids at lower dimensional subspaces. Coarse scale features are stable and hence more generic in terms of defining rules for recognizing patterns. Fine scale features are less stable but still play important role in recognizing specific properties of patterns. Thus coarse scale grids result in defining more generic rules, which are suitable for 'small sample size' case studies, and fine scale grids result in defining more specialized rules, which are good for enhancing classification accuracy. We use this strategy in our proposed approach for recognizing patterns in multidimensional data.

An example of projecting a given multidimensional data set to two dimensional subspaces is shown in Figure 1. The given data is projected to a two dimensional subspace in such a way so that it forms clusters which are spread out in the two dimensional subspace. This subspace is divided into coarse grids where one or more grids may cover a cluster as shown in Figure 1a. A coarse grid may have data belonging to a single class, in which case the grid forms a generic rule

for that class. In situations where a grid has data belonging to more than one class, such grids are further divided into fine grids as shown in Figure 1b (four coarse grids are shown to be divided into fine grids). Fine grids are used to add premises to generic rules which result in specialized rules, and these rules suit situations where separation of data belonging to different classes is difficult at coarse level. We further propose that multi-class data in a single grid can be discriminated with the aid of those fine grids which are obtained by projecting the multidimensional data to a different lower dimensional subspace where projected data is fully spread out as shown in Figure1c. The spread out data in this lower dimensional subspace is divided into fine grids, and such fine grids are added as premises to the generic rules obtained from the coarse grids of Figure1a. This results in specialized rules which improve classification accuracy.

Keeping the above reasoning in view, we propose strategy for recognition of patterns in multidimensional data that is carried out in four steps: i) Projecting a multidimensional data set to lower dimensional spaces, and ii) Creating multi-scale grids at lower dimensional spaces iii) Defining generic rules using coarse scale grids iv) Defining specialized rules using fine scale grids. The rules obtained in steps iii) and iv) are used for recognizing patterns.

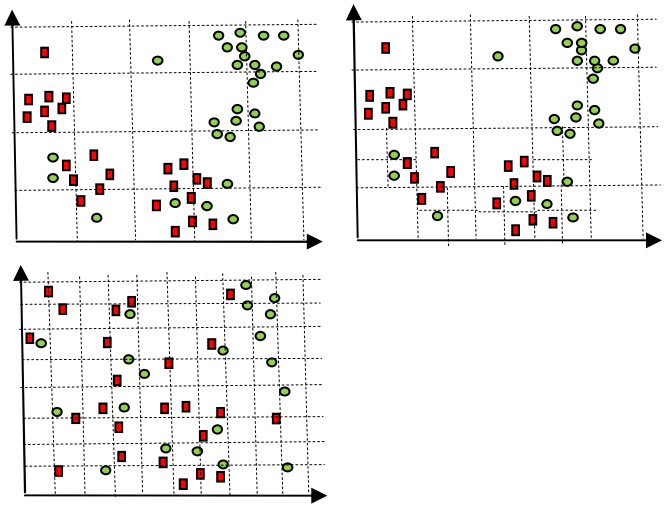

Figure1(a) Multidimensional data projection to a 2D space where coarse grids separate the clusters of data belonging to two classes (b) shows four coarse grids having data of more than one class divided into fine grids (c) shows spread out projection to a 2D space divided into fine grids.

A. Multidimensional Data Projections

Multidimensional data is processed row by row and each row is projected along four projection vectors. Two of the four projection vectors are defined by principal component analysis and the rest two projection vectors are defined by multiple discriminant analysis. Three lower dimensional (i.e. 2D) subspaces are created with the help of these four projection vectors. The three two dimensional subspaces should be defined in such a way that it facilitates construction of coarse and fine grids. The first two dimensional subspace will have both vectors from principal component analysis. As principal component analysis attempts to spread the projected data, the resulting two dimensional subspace will suit for creating fine grids. More details on principal component analysis are given below. The second two dimensional subspace will have both vectors from multiple discriminant analysis. Here also, as multiple discriminant analysis attempts to spread the projected data using class information, the resulting two dimensional subspace will suit for creating fine grids. More details on multiple discriminant analysis are given below. The third two dimensional subspace will have one vector from principal component analysis and the second vector from multiple discriminant analysis. As the principal component analysis attempts to spread the projected data without using class information while as multiple discriminant analysis uses class information, the resulting two dimensional subspace will have data spread in clusters where each cluster will tend to have data mostly belonging to one class only. This projection will suit for creating coarse grids. This transforms multidimensional data to three two dimensional subspaces which are divided into coarse and fine grids for pattern recognition.

i) Projection With Principal Component Analysis

Principal component analysis (PCA) is a widely-used statistical technique and it best represents the data in a least-squares sense. It works by replacing the original (numerical) variables with new numerical variables called "Principal Components". PCA captures the most descriptive features with respect to packing most "energy". This involves minimizing the criterion function $j_{d'}$ for a d'-dimensional projection:

$$j_{d'} = \sum_{k=1}^{n} \left\| (\mathrm{m} + \sum_{i=1}^{d'} a_i \mathrm{e}_i) - \mathrm{x}_k \right\|^2$$

where $x_1,\ldots x_n$ are n data points to be projected to a low dimensional space, m is the data mean, a_k are coefficients that minimize the criterion function, vectors $e_1,\ldots..e_d$' are the d' eigenvectors of the scatter matrix having the largest eigen values.

ii) Projection With Multiple Discriminant Analysis

Fisher linear discriminant analysis (FDA) is a simple algorithm that best separates the data in a least-squares sense. It is used for both dimension reduction and classification. In either case, FDA attempts to minimize the Bayes error by selecting the most discriminant feature vectors. To increase the effective dimension of the projected space the use of Multiple Discriminant Analysis (MDA) instead of FDA is used.

Multiple discriminant analysis adopts a perspective similar to Principal Components Analysis, but PCA and MDA are mathematically different in what they are maximizing. MDA maximizes the difference between values of the dependent, whereas PCA maximizes the variance in all the variables accounted for by the factor. A technique that extracts invariant but descriptive features involves maximization of the criterion function given below:

$$J(v) = \frac{|W^t S_B W|}{|W^t S_W W|}$$

where W is the weight vector of a linear feature extractor and S_B and S_W are symmetric matrices designed such that they measure the desired information and the undesired noise along the direction W. S_B measures the separability of class centers (between-class variance), and S_W measures the within-class variance. S_B and S_W are given by:

$$S_B = \sum_{j=1}^{C} N_j.(m_j - m)(m_j - m)^T$$

$$S_W = \sum_{j=1}^{C} \sum_{i=1}^{N_j} (x_i^{(j)} - m_j)(x_i^{(j)} - m_j)^T$$

where $\{ x_i^{(j)}$, i=1,…,$N_j\}$, j=1,…C are feature vectors of training samples, C is the number of classes, N_j is the number of the samples of the jth class, $x_i^{(j)}$ is the ith sample from the jth class, m_j is mean vector of the jth class, and m is grand mean of all examples.

PCA and MDA, each has its own pros and cons. MDA deals directly with discrimination between classes, whereas PCA does not pay particular attention to the underlying class structure. When the data of each class can be represented by a single Gaussian distribution and share a common covariance matrix, MDA will outperform PCA. By contrast, when the number of samples per class is small or when the training data non-uniformly sample the underlying distribution, PCA might outperform MDA.

PCA and MDA algorithms were used to project the data to a feature vector space. Each algorithm used two eigenvectors that corresponded to the two largest distinct eigen values for defining the axes of feature vector space.

B. Rule Extraction Using Coarse and Fine Grids

The projections to two dimensional subspaces are divided into a number of cells or subspace grids. A subspace grid can form a premise of a rule. Rules are extracted by considering subspace grids as its possible premises. The rule extraction process is summarized below:

The core procedure of the algorithm is a recursive process to form a decision tree from the current set of subspace grids. Let *S* be a set of instances at a node and if all the instances in S belong to the same class, then that node is labeled with the class name of those instances. Otherwise, *S* contains representatives of more than one class. A vector is selected to partition *S* into subsets $S_1, S_2, S_3, ...S_n$ where S_i contains those members of *S* that have ith subspace grid along the selected vector. Let *A* be a set of m vectors $\{A_1, A_2, A_3, ...,A_m\}$, *C* a set of p classes $\{C_1, C_2, C_3, ...C_p\}$. The set of possible values of a vector A_i (for forming subspace grids) is referred to as Range (*Ai*). Each example in *S* is an m+1 tuple of the form ($V_1, V_2, V_3, ..., V_m, C_k$) where V_i ε Range(A_i), i =1,….m and C_k ε *C* is the class of that example. The probability of occurrence of examples of class C_k in a set S_l, $P_{Sl,Ck}$, is the proportion of examples in S_l that are in class C_k. The information measure, which gives a measure of randomness of example distribution in S_l over the possible classes in *C*, is given by

$$E(A_i, S) = \sum_{V_l \varepsilon Range(A_i)} I(S_l) \frac{n(S_l)}{n(S)}$$

The algorithm aims to partition *S* to produce subsets S_l in which the examples are distributed less randomly over possible classes. To choose the vector that would best achieve this, the algorithm partitions *S* into subsets S_l corresponding to values V_l of a vector A_i. If the number of examples containing value V_l in *S* is n(*S*) and the number of examples containing value V_l in subset S_l is n(S_l), then information entropy of the resulting partition is given by:

$$I(S_l) = -\sum P_{S_l,C_k} Log_2 P_{S_l,C_k}$$

The algorithm chooses that vector A_i for branching which maximizes the following quantity:

Gain (A_i, *S*) = I(*S*) – E(A_i, *S*)

If Gain(A_i, *S*) is same for more than one vector, then one of them is chosen randomly.

IV. RESULTS AND DISCUSSION

The results of the proposed approach are demonstrated on IRIS data set. The IRIS data set [10] is a collection of continuous-valued data commonly used in bench marking pattern classification algorithms. Each example in the set is described in terms of four numerical attributes: Sepal_length, Sepal_width, Petal_length, Petal_width and can be classified into one of three categories, Iris_Setosa, Iris_Versicolor or Iris_Virginica. The total number of examples is 150. In this application, 100 examples were randomly picked for extracting rules and all the 150 examples were used for testing the extracted rules.

The IRIS data set was projected to three two dimensional subspaces. The first subspace used one vector from PCA

and second vector from MDA corresponding to their highest Eigen values. The subspace is shown in Figure2a. The subspace was divided into 5 by 5 coarse grids as shown in Figure2b. The rules extracted from this coarse grid are shown below:

Rule 1: IF x-axis lies between 2.286 and 3.69
AND y-axis lies between 0.986 1.939
THEN class is 1 with probability of 1
Rule 2: IF x-axis lies between 3.690 5.094
AND y-axis lies between -0.921 0.032
THEN class is 2 with probability of 1
Rule 3: IF x-axis lies between 5.094 6.498
AND y-axis lies between -0.921 0.032
THEN class is 2 with probability of 1
Rule 4: IF x-axis lies between 6.498 7.902
AND y-axis lies between -2.828 -1.875
THEN class is 3 with probability of 1
Rule 5: IF x-axis lies between 7.902 9.306
AND y-axis lies between -2.828 -1.875
THEN class is 3 with probability of 1
Rule 6: IF x-axis lies between 7.902 9.306
AND y-axis lies between -1.875 -0.921
THEN class is 3 with probability of 1
Rule 7: IF x-axis lies between 5.094 6.498
AND y-axis lies between -1.875 -0.921
THEN class is 2 with probability of 0.947
Rule 8: IF x-axis lies between 6.498 7.902
AND y-axis lies between -1.875 -0.921
THEN class is 3 with probability of 0.629

The above coarse rules classify 132 examples out of 150 examples correctly giving a classification accuracy of 88%. The Rule 7 above corresponds to a grid that has examples belonging to two classes: class3 and class2. The proportion of examples belonging to class2 (i.e 18/19=0.947) and class3 (i.e. 1/19) is the probability with which this rule can classify a given example as belonging to class2 or class3 respectively. Similarly, the Rule 8 above corresponds to a grid that has examples belonging to two classes: class3 and class2. The proportion of examples belonging to class3 (i.e 17/27=0.629) and class2 (i.e. 10/27) is the probability with which this rule will classify a given example as belonging to class3 or class2 respectively.

Thus the grids corresponding to Rule 7 and Rule 8 have data belonging to more than one class and its use will result in classification error. To address this issue, the grids corresponding to Rule 7 and Rule 8 are divided into fine grids (each grid is divided into four grids). Rule 7 and Rule 8 are redefined by using fine grids. Note that it is not necessary that all new fine grids will have data in it, but those that do will give rise to redefined rules. The 8 new fine grids give rise to the following 4 rules:

Rule 7: IF x-axis lies between 5.796 6.498
AND y-axis lies between -1.398 -0.921
THEN class is 2 with probability of 1
Rule 8: IF x-axis lies between 6.498 7.20
AND y-axis lies between -1.398 -0.921
THEN class is 2 with probability of 1
Rule 9: IF x-axis lies between 6.498 7.20
AND y-axis lies between -1.875 -1.398
THEN class is 3 with probability of 0.714
Rule 10: IF x-axis lies between 5.796 6.498
AND y-axis lies between -1.875 -1.398
THEN class is 2 with probability of 0.5

The above set of rules classify 145 examples out of 150 examples correctly giving a classification accuracy of 96.7%. The Rule 9 above corresponds to a grid that has examples belonging to two classes: class3 and class2. The proportion of examples belonging to class3 (i.e 10/14=0.714) and class2 (i.e. 4/14) is the probability with which this rule will classify a given example as belonging to class3 or class2 respectively. Similarly, the Rule 10 above corresponds to a grid that has examples belonging to two classes: class3 and class2. The proportion of examples belonging to class3 (i.e 1/2 = 0.5) and class2 (i.e.1/2 = 0.5) is the probability with which this rule will classify a given example as belonging to class3 or class2 respectively.

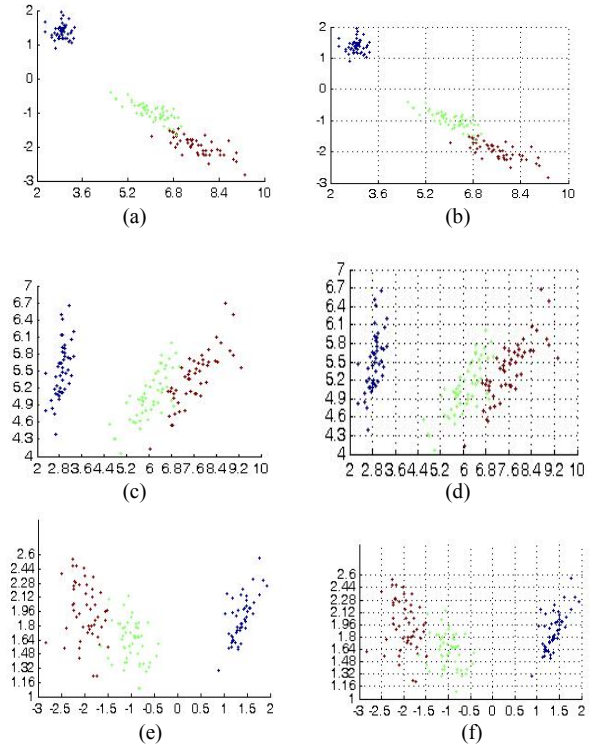

Figure 2.(a,b) Subspace and grids using PCA and MDA algorithm (c, d) Subspace and grids using PCA algorithm, (e,f) Subspace and grids using MDA algorithm.

Thus the grids corresponding to Rule 9 and Rule 10 have data belonging to more than one class and its use will result in classification error. To improve the results further, Rule 9 and Rule 10 are refined by adding new premises. The new premises are obtained by using fine grids (10 by 10 grids) of other two projected subspaces: subspace 2 (using vectors from PCA algorithm) (Figure2c,d) and subspace 3 (using vectors from MDA algorithm) (Figure2e,f). Rule 9 gets modified as below:

Rule 9a: IF x-axis lies between 6.498 7.20
AND y-axis lies between -1.875 -1.398
THEN class is 3 with probability of 0.714
Rule 9b: (IF x-axis lies between 6.498 7.20
AND y-axis lies between -1.875 -1.398)
from subspace 3
AND
(IF x-axis lies between -1.875 -1.398
AND y-axis lies between 1.394 1.540)
from subspace 2
THEN class is 2 with probability of 1
Rule 9c: (IF x-axis lies between 6.498 7.20
AND y-axis lies between -1.875 -1.398)
from subspace 3
AND
(IF x-axis lies between 6.498 7.20
AND y-axis lies between 5.368 5.632)
from subspace 1
THEN class is 2 with probability of 0.83

Similarly, Rule 10 gets modified as below:
Rule 10a: IF x-axis lies between 5.796 6.498
AND y-axis lies between -1.875 -1.398
THEN class is 2 with probability of 0.5
Rule 10b: (IF x-axis lies between 5.796 6.498
AND y-axis lies between -1.875 -1.398)
from subspace 3
AND
(IF x-axis lies between -1.875 -1.398
AND y-axis lies between 1.540 1.687)
from subspace 2
THEN class is 3 with probability of 0.75

The above set of rules classify 148 examples out of 150 examples correctly giving a classification accuracy of 98.67%. The classification accuracy results are summarized in Table 1 below:

TABLE I CLASSIFICATION ACCURACY WITH RULES FROM GRIDS

Rules Used	Classification accuracy
Rules from only Coarse Grids of subspace 3	88%
Rules from Coarse and Fine Grids of subspace 3	96.7%
Rules from Coarse and Fine Grids of subspace 3 With premises supplemented from subspaces 1 and 2.	98.67%

V. CONCLUSION

This paper introduced coarse and fine subspace grids to recognize patterns in multidimensional data. Multidimensional data was projected to three subspaces. The PCA and MDA algorithms were used to define lower dimensional subspaces. Coarse and fine grids were obtained from the lower dimensional subspaces by dividing the range of values associated with each vector of a subspace into equal number of parts. A recursive procedure was employed to extract rules from subspace grids. The extracted set of rules was used for recognition of patterns present in the multidimensional data. The approach was tested on the bench mark IRIS data set. The paper showed that the use of subspace grids to recognize patterns in multidimensional data produced good results.

REFERENCES

[1] H. Chai, H. and D. Domeniconi, "An Evaluation of Gene Selection Methods for Multi-class Microarray Data Classification", Proceedings of the Second European Workshop on Data Mining and Text Mining in Bioinformatics. pp. 3-10, 2004.

[2] G. Hori, M. Inoue, S. Nishimura, and H. Nakahara, "Blind gene classification based on ICA of microarray data", Proceedings of 3rd International Conference on Independent Component Analysis and Blind Signal Separation. San Diego. pp. 332-336, 2001.

[3] R. Pique-Regi1, A. Ortega, and S. Asgharzadeh, "Sequential Diagonal Linear Discriminant Analysis (SeqDLDA) for Microarray Classification and Gene Identification", Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference–Workshop. pp. 112–116, 2005.

[4] H. Xiong, Y. Zhang, and X. Chen, "Data-dependent Kernel Machines for Microarray Data Classification", IEEE/ACM Trans. Comput. Biology Bioinform. pp. 583-595, 2007.

[5] L. Wang, J. Zhu, and H. Zou, "Hybrid Huberized Support Vector Machines for Microarray Classification", Proceedings of the 24 th International Conference on Machine Learning. Corvallis, Oregon, pp 983-990, 2007.

[6] K. Kim, and S. Cho, "Ensemble classifiers based on correlation analysis for DNA microarray classification", Neurocomputing . 70: pp. 187–199, 2006.

[7] S. M. Hosseyninia, F. Roosta, A. A. S. Baboli, G. R. Rad, "Improving the performance of MPCA+MDA for Face Recognition", 29th Iranian Conference on Electrical Engineering, pp. 1-5, 2011.

[8] M. Arif Wani, “Incremental Hybrid Approach for Microarray Classification”, Proceedings of the Seventh International Conference on Machine Learning and Applications, San Diego, USA, pp. 514-520 , December, 2008.

[9] M. Arif Wani, "Microarray Classification using Subspace Grids", Proceedings of the Tenth International Conference on Machine Learning and Applications, Hawaii, USA, Volume 1, pp. 389-394 , December, 2011.

[10] R. A. Fisher, “The use of multiple measurements in taxonomic problems”, Ann. of Eugenics, 7, 179-188, 1936.