Global Journal of Computational Intelligence Research.
ISSN 2249-0000 Volume 4, Number 1 (2014), pp. 27-44

http://www.ripublication.com

Adaptive Biclustering with VQ–INC–EXT for Microarray Gene Expression Data

Ms. S. Anandhavalli[1] and Dr. S.K. Srivatsa[2]

[1]Research Scholar, Jawaharlal Nehru Technological University, Hyderabad-500085,India
[1]Associate Professor, Department of M.C.A., St. Joseph's College of Engineering, Chennai–600 119, India.
[2]Senior Professor, St. Joseph's College of Engineering, Chennai-600119, India
E-mail: [1]anandhi.bharat@gmail.com, [2]profsks@rediffmail.com

Abstract

In this paper, we determine the analysis and classification of genes and their phenotypes in RNA expression levels by considering the influence of vigilance parameter in FLEXFIS. Since the classification problems are interrelated, we want to find "marker genes" that are differentially expressed in particular sets of "conditions." We have developed a method that simultaneously clusters genes and conditions, finding distinctive clusters with less number of rules generated. In a cancer context, these representations correspond to genes that are markedly up or down regulated in patients with particular types of tumors. Our method is such that structures in matrices of expression data can be found in eigenvectors in particular the singular value decomposition (SVD), with closely integrated normalization steps, corresponding to the characteristic expression patterns across genes or conditions. A modified version of vector quantization is used in Flexible Fuzzy Inference System (FLEXFIS) specifically for Takagi Sugeno fuzzy model to identify the appropriate cluster. It adopts a single pass incremental learning approach for the antecedent parts of the rules' learning process while eliminating the outliers and the problem of cluster projection. An evaluation is done on the microarray gene expression data. A comparative study of the performance analysis for both the conventional and incremental version of vector quantization is also presented in this paper.

Keywords: Vigilance parameter, FLEXFIS, Takagi Sugeno Fuzzy Model, vector quantization, microarray gene expression data.

1. Introduction

1.1 FLEXIBLE Fuzzy Inference System (FLEXFIS)

The development of a successful fuzzy model for a chaotic system that exhibits a nonlinear dynamic behavior, with the identification of consequent parameters is an important yet difficult task which is traditionally tackled by trial and error process. FLEXFIS finds a reasonable connection between the adaptation of nonlinear premise parameters and linear consequent parameters. It uses the modified version of vector quantization to eliminate the outliers that fall in the less dense region of the feature space.

1.2 Takagi-Sugeno Fuzzy model (TSK)

The zero order Takagi-Sugeno fuzzy system model can approximate any real occurring non linear relationship to a certain degree of accuracy. The incremental learning for evolving Takagi-Sugeno fuzzy model is done by connecting premise and rule learning. This has been accomplished by updating the premise part and adapting the consequent parameter of the rule with respect to the winning cluster. It helps in constructing fuzzy inference system based on given data or knowledge of the human experts.

1.3 Vector Quantization

Vector quantization encodes large set of training data into a small set of representative points, thus achieving compression in representing the data. Such compression of data in genes and their phenotypes in RNA expression levels results in overlapping clustering of incoming data points. However, the modified version of vector quantization, viz., vector quantization incremental extended version eliminates the outliers while testing the quality of data in incremental learning in Takagi-Sugeno Fuzzy model. It has the ability to build clusters in incremental manner without preparameterizing the number of clusters. The distance of the new data point to the surface of the multidimensional ellipsoid spanned by a cluster is calculated and the surface can be updated synchronously to the cluster center.

2. Literature Survey

2.1 Vector Quantization (VQ)

Fuzzy systems are widely used in areas such as pattern recognition, classification, fault detection, control [12], automation and identification of tasks. Various approaches have been used to define linguistic interpretation models [11] and rule base simplification. A fuzzy modeling procedure comprises of two stages, viz., structure identification and parameter identification. [2]

The vector quantization is a clustering technique which is used to partition the input space. The vector quantization is exploited in association with the vigilance parameter. [10] It is characterized by the ability to build up clusters in incremental manner. It calculates the range of influence of clusters in each direction, finds the nearest winning cluster by calculating the distance from a new data point to the surface (instead of centers as in conventional VQ) of already obtained clusters. This prevents the generation of a new cluster very close to or even inside already existing ones. [1]

Vigilance parameter has been used to update cluster with each new incoming data point and generation of new cluster. It steers the tradeoff between generating new clusters, rules and updating already existing ones. Already generated clusters are moved in local areas bounded by the vigilance parameter ρ. Thus plasticity and stability is ensured by adapting to new information without changing any already learned clusters. Therefore it overcomes plasticity-stability dilemma. The adaptive vigilance parameter for the clustering prevents an incorrect cluster partition due to an inappropriate apriori setting of ρ.

2.2 Curse of dimensionality

The dependency of the vigilance parameter ρ on the (p+1) dimensional space can be explained with the curse of dimensionality. According to the curse of dimensionality, the higher the dimensions are the greater the distance between two adjacent data points would be. When the value of vigilance parameter is larger, the algorithm prevents the generation of too many clusters causing strong over fitting effects.

2.3 Drawback

Vector quantization cannot be reasonably applied for online processes, where incremental clustering is demanded. Clustering techniques update their parameter for each newly loaded data block or even for each single data sample without taking into account prior data. This is because it iterates over the loaded data buffer several times. If this would be carried out for each incremental learning step i.e., for each actual loaded data block separately, the cluster centers would only represent a reliable partition of this data block and forget the older data completely Moreover, the number of clusters has to be known in advance, which can be a significant drawback, especially in case of high-dimensional data sets as the number of clusters cannot be seen. Furthermore, in the case of online clustering the number of clusters are never known in advance as the data have to be processed through the algorithm as these are loaded or recorded.

Vector quantization generates over clustering i.e., two new small clusters for data points lying near the range of influence of big cluster. Computation of Euclidean distance of the new incoming point to the nearest cluster center and comparing this distance with the threshold ρ can result in a wide data cloud. A new data point can lie near or even inside the range of influence, but still be far away from the center.

(i) **(ii)**

Fig. 2.1: Drawback of VQ.

In this fig 2.1, two clusters overlap inside the range of influence of another big cluster, thus resulting in over clustering. This drawback can be overcome by an adaptive vigilance parameter ρ that defines the range of influence of the nearest cluster. It eliminates overclustering that is formed in (i) thereby increasing the surface of the big cluster as shown in (ii). The surface of the big cluster adapts to some of the data points that appear in the overclustering. This is a quite sophisticated issue as it depends on the position of the current sample relative to the cluster ellipsoid.

Hence, the distance of the new data point to the surface of the multidimensional ellipsoid spanned by a cluster is calculated and the surface can be updated synchronously to the cluster center. The conventional vector quantization, with the conventional winning strategy generates two new small clusters for data points lying near the range of influence of the big cluster; but the VQ-INC-EXT extends the surface of this big cluster. The distance along the direction from the current point towards the cluster center is taken as an approximation of the shortest distance to the surface,

The local learning approach is necessary for a complete incremental learning of the system, as it triggers higher flexibility for adjoining rules when new operating conditions occur. The adaptation formulation of local learning results in weighted least squares that calculate a new update for the linear parameter c_1 each time a new data point comes in. Such conditions should be incorporated into the system, which is shown in the fig 2.2.

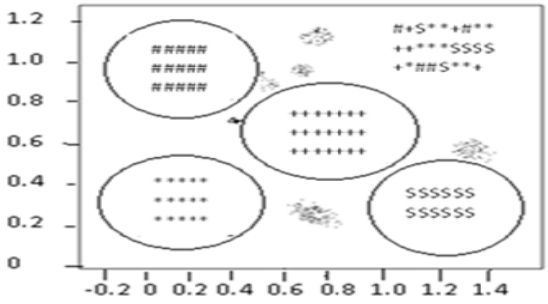

Fig. 2.2: Incorrect model with data cloud.

The adaptation process to extend the already generated models has four clusters represented with the identification of microarray gene data's core parameters viz., acute Leukemia (B–and T-cell acute lymphocytic leukemia [ALL] and acute myelogenous leukemia [AML]. In addition, we also find another collection of all these parameters formed as a separate cloud towards the rightmost upper corner of the diagram. This separate cloud is treated as a bad cloud that results in over clustering. Hence, such outliers forming part of the bad cloud have to be eliminated by way of using the modified version of extended incremental vector quantization. It doesn't result in the entire elimination of the incoming data; instead those incoming data points have to be moved to their respective clusters by calculating the distance between such data points to their respective clusters. In order to overcome this deficiency a connection of recursive weighted least squares with an incremental premise parameter and rule evolution is used. The evolution of premise parameter and rule learning can be done with the usage of clustering the input/output space into local parts and projection of these clusters to form the premise parts of the rules. Hence Extended Incremental version of Vector Quantization in association with the vigilance parameter is used for an update of cluster surface with new incoming data point. It is necessary to exclude outliers or other faulty points from the update process, as this spoils the data space and the clustering process.

In section 3, we discuss about our proposed system. Section 4 discusses with the performance analysis of the work by considering two factors viz., misclassification rate and vigilance parameter. Section 5 deals with the conclusion and future work.

3. Proposed System

The modified version of vector quantization viz., extended incremental version of Vector Quantization (VQ-INC-EXT) is used to eliminate over clustering which was present in conventional Vector Quantization (VQ). The performance of adaptive vigilance parameter ρ is evaluated in the proposed system. The microarray gene expression data, when given as input to the Fuzzy Inference Engine, helps in checking the progress of gene expressions. The approach followed to find the predicted values of the parameters during the learning process of the vigilance parameter is explained with the help of our algorithm viz., Evaluate_Vigilance that takes as input a set of data points and produces as output a set of predicted values in order to decide about the cluster identification process (normal and abnormal) on a particular instant of action, is mentioned in what follows

Evaluate_vigilance Algorithm
Input : Data Set
Output : Predicted Values

STEPS

1. Extract the distribution of patterns in the form of various parameters in the feature space

2. Divide the input space into smaller local regions, identified as clusters.
3. Update the training set as and when new data points are derived.
4. Characterize the local properties by normalizing the membership functions in the fuzzy set.
5. Use sigmoidal threshold to characterize the global properties of the learning process.
6. Use adaptation of vigilance parameter for cluster updates.
7. Fine tune the vigilance parameter by fixing the threshold value

In particular, in analyzing microarray cancer data sets we wish to identify both clusters of genes that participate in common regulatory networks as well as clusters of experimental conditions associated with the effects of these genes, viz., clusters of cancer subtypes. In both cases we use similarities between expression level patterns to determine clusters. Clearly, prior knowledge of clusters of genes can help in clustering experimental conditions, and vice versa. In the absence of knowledge of gene and condition classes, it would be useful to develop partitioning algorithms that find latent classes by exploiting relations between genes and conditions. Partitioning clustering methods can be used to identify a particular cluster to which the incoming data belong. However, the influence of a particular gene with vigilance parameter value contributes to the overall shape of the cluster. The data set with abnormal behaviour are given as input and the Fuzzy Inference system is trained in an incremental manner. Such training helps the system to adapt to abnormal data sets that are encountered. The adaptive vigilance parameter has increased the accuracy of cluster partitioning.

3.1 Extended Incremental Version of Vector Quantization (VQ-INC-EXT)

The VQ-INC-EXT algorithm finds the nearest winning cluster by calculating the distance from a new data point to the surface of already obtained clusters. Clusters are built up in incremental manner. The risk of approximation of the noise, identification and elimination of such outliers is high and over-fitting effect is low.

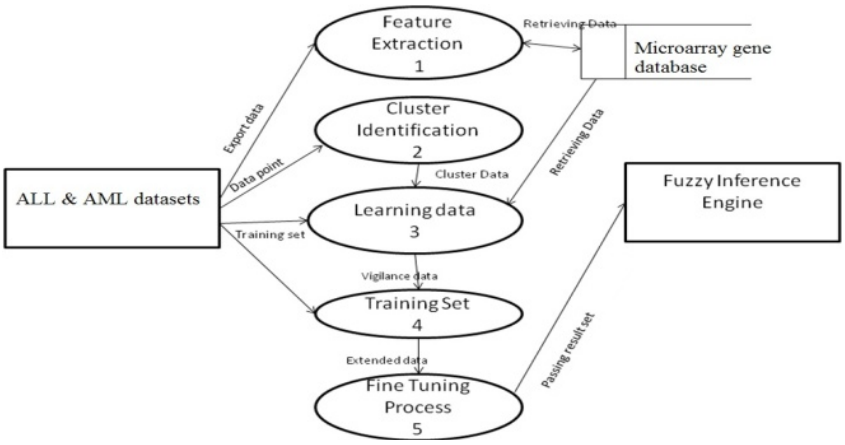

Fig. 3.1: Module Descriptions.

It gives better result when high noise is present in the data. It generates too few clusters for highly nonlinear approximation cases since it forms a big single cluster of data, considering the presence of similarities in the data.

The representation of the modules specifications for microarray gene datasets is given in figure 3.1.

The input to the experimental study is the data set derived from the microarray gene database and the output is the set of predicted values stored in the inference engine.

3.2 Feature Extraction

Feature extraction is the environment where the distribution of patterns in feature space changes with respect to time. It is necessary to employ an adaptive learning algorithm which makes them suitable in well-known classical feature extraction and projection approaches. They have enormous promise in such areas as revealing function of genes in various cell populations, tumor classification, drug target identification, understanding cellular pathways, and prediction of outcome to therapy. A value in the matrix A_{ij} could either represent absolute expression levels (such as from Affymetrix Gene Chips) or relative expression ratios (such as from cDNA microarrays) depending on the type of chip technology used,

A specific assumption in tumor classification is that

samples drawn from a population containing several tumor types have similar expression profiles if they belong to the same type. Observing several experiments, each of which has multiple tumor types, gives an assumption; for tumors of the same type there exist subsets of overexpressed or under expressed genes that are not similarly overexpressed or under expressed in another tumor type.

Under this assumption, the matrix *A* could be organized in a checkerboard-like structure with blocks of high-expression levels and low-expression levels. A block of high-expression levels corresponds to a subset of genes (subset of rows) that are highly expressed in all samples of a given tumor type (subset of columns). However, this simple checkerboard-like structure can be confounded by a number of effects. In particular, different overall expression levels of genes across all experimental conditions or of samples across all genes in multiple tumor datasets can obscure the block structure. Consequently, rescaling and normalizing both the gene and sample dimensions could improve the clustering and reveal existing latent variables in both the gene and tumor dimensions. The SVD has been applied to microarray experiment analysis to find underlying temporal and tumor patterns. The two different microarray gene datasets viz., ALL and AML and AD and NL are considered. The AD and NL dataset contains 675 genes and 156 samples. Among these 156 samples, 1 to 139 samples belong to the AD class type and 140 to 156 samples belong to the NL class type. The ALL and AML dataset is of dimension 7192x38. In this data set 1 to 26 gene column data are ALL and 26 to 38 gene column data are AML. The subtractive clustering method is used for partitioning the data in the input space.

The raw data in many cancer gene-expression datasets can be arranged in a checkerboard matrix structure form as schematized in Figure 3.1

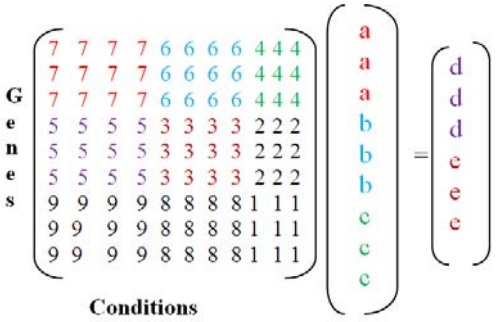

Fig. 3.1: Checker board structure.

The vigilance parameter, named ρ, has considerable influence in the classification, the higher is the vigilance parameters and the more accurate is the classification. Table 3.1 shows the sample leukemia data set that are used as the input data for the feature extraction.

Table 3.1: Sample data set.

Gene description	Accession Number	Call	Endogenous Control	Affi_bio
-831	-934	-471	-1003	-1001
-653	-577	-490	-761	-520
-462	-214	-184	-541	-163
75	142	32	109	-38
381	271	213	435	281
-118	-107	1	-129	-137
-565	-101	-260	-399	-247

3.3 Cluster Identification

Cluster identification is a process of identification of the winning cluster to which the incoming data point belongs. Multi-layer perceptron is used for the purpose of identification of the cluster because it exhibits the nonlinear behavior of various patterns. The Leukemia Dataset viz., acute Leukemia (B–and T-cell acute lymphocytic leukemia [ALL] and acute myelogenous leukemia [AML] are considered as parameters for deriving the test data. The patient distributions of the different diseases of the leukemia dataset become separated in the two-dimensional graphs generated by projecting the expression profiles onto the gene class partition vectors of the biclustering method as shown in Fig 3.2

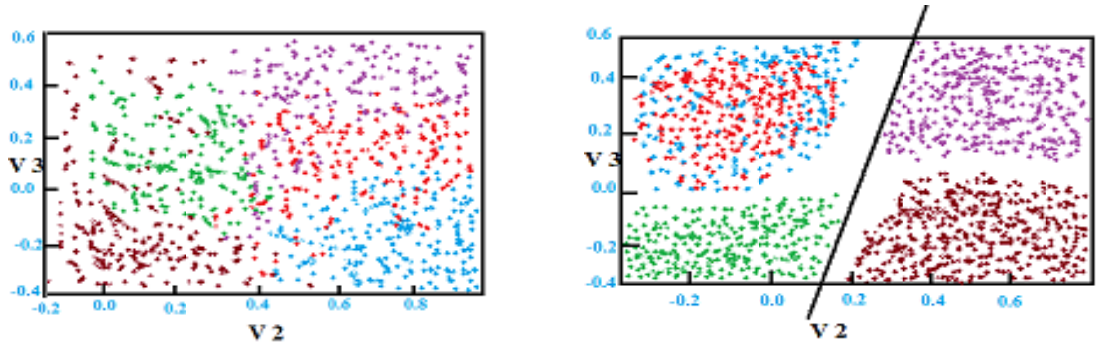

Fig. 3.2: Cluster Identification.

B-cell ALL samples are denoted by red dots, T-cell ALL by blue dots, and AML by green dots. The AD and NL datasets are denoted by pink dots and maroon dots respectively. In this analysis we preselected all genes that had positive Affymetrix average difference expression levels. The bistochastic method also partitions the patients well, with only one ambiguous case that is close to the boundary between ALL and AML. Moreover, biclustering does not require specification of the number of desired clusters or lengthy searches for subsets of genes. These parameters are used for the cluster partitioning wherein the incoming data point viz., a new data, can be placed accordingly in a particular cluster based on the partitions of the input space. The calculations to obtain the interactions are simpler than bistochastization, as they are done by a simple formula from the study of two way ANOVA with no iteration. Therefore we do not automatically discard them as was done in the previously discussed normalizations. Such calculation is compared with the vigilance parameter which helps to predict the presence of cancerous cell. It also helps to train the Fuzzy Inference System for the abnormal conditions that has been encountered. The adaptive vigilance parameter for the clustering prevents an incorrect cluster partition due to an inappropriate a priori setting of ρ

If the incoming data with respect to a particular microarray gene data is the first data point, it is considered as the data point and the cluster. If the incoming data lies inside the range of influence of any cluster, the distance of the selected data point to those cluster surfaces is calculated by using Euclidean distance. The winning cluster is identified by taking the minimum of all the calculated distance. If the minimum distance is zero no new clusters are added. Otherwise, if the current data point lies outside of all clusters' range of influence, the minimum distance of the new data point to the surface of all the clusters is calculated. The positioning of the incoming data is done based on the distance value. Further partitioning of the ALL cases is obtained by applying a normalized cuts clustering method to the biclustering Eigenvectors, and produces a clear separation between T-cell and B-cell ALL.A common and useful practice in microarray analysis is transforming the data by taking logarithms. The resulting transformed data typically have better distributional properties than the data on the original scale — distributions are closer to normal, scatterplots are more informative. The log interactions normalization method begins by calculating the

logarithm L_{ij} = log (A_{ij}) of the given expression data and then extracting the interactions between the genes and the conditions, where the term "interaction" is used as in the analysis of variance (ANOVA).

As above, the log-interactions normalization is motivated by the idea that two genes whose expression profiles differ only by a multiplicative constant of proportionality are really behaving in the same way, and we would like these genes to cluster together. In other words, after taking logs, we would like to consider two genes whose expression profiles differ by an additive constant to be equivalent. This suggests subtracting a constant from each row so that the row means each become 0, in which case the expression profiles of two genes that we would like to consider equivalent actually become the same. Similar is the case for the conditions (columns of the matrix). Constant differences in the log expression profiles between two conditions are considered unimportant. We subtract a constant from each column so that the column means become 0.

Let the the average of the i^{th} row be

$L_i = (1/m) \_ m_j=1$

the average of the j^{th} column

$L_j = (1/n) \_ n_i =1$

the average of the whole matrix,

$L_{ij} = (1/mn) \_ n_i =1 \_ m_j=1$

the result of these adjustments is a matrix of interactions K = (Kij) such that

$K_{ij} = L_{ij}\_L_i.\_L_j + L...$

It turns out that these adjustments to the rows and columns of the matrix to achieve row and column means of zero can all be done simultaneously by a simple formula as stated in Fig 3.3.

The interaction K_{ij} between gene *i* and condition *j* captures the extra (log) expression of gene *i* in condition *j* that is not explained simply by an overall difference between gene *i* and other genes or between condition *j* and other conditions, but rather is special to the combination of gene *i* with condition j.

Again, as described before, we apply the singular value decomposition (SVD) to the matrix *K* to reveal block structure in the interactions. There are two clusters already existing hence the incoming data may belong to any one of the cluster if not the system will be trained for the new incoming data.

3.4 Training set

Takagi-Sugeno fuzzy model recursively updates the structure of the model based on the potential of the input. A new rule is added when the distance from the new data point to the winning cluster is greater than the vigilance parameter. A rule is modified when the distance from the new data point to the winning cluster is lesser than the vigilance parameter. The condition for the generation of the new rule is shown in fig 3.4

$$\|\vec{x} - \vec{c}_{\text{win}}\|_A \geq \rho \quad \text{and} \quad \vec{x} \text{ is not faulty}$$

$$\rho = \text{fac} \frac{\sqrt{p+1}}{\sqrt{2}}$$

Fig 3.4: Generation of new cluster (Rules)

The training set contains new datasets. These sample training set are shown in table 3.2. The clusters are incrementally updated and generated for creating a dynamically changing and evolving classifier. A new rule will be generated whenever a cluster is being formed for the incoming data.

Table 3.2: Training set.

Gene description	Accession number	call	Endo–genous control	Affi_bio
-41	363	155	-115	361
-831	-934	-471	-1003	-1001
-653	-577	-490	-761	-520
-462	-214	-184	-541	-163
75	142	32	109	-38

These rules will be stored in Fuzzy Inference Engine in If-Then format. They are formed based on clusters of genes that participate in common regulatory networks and clusters of experimental conditions associated with clusters of cancer subtypes. In both cases we want to use similarities between expression level patterns to determine clusters.

Table 3.3: Rule set

RULE1 IF (N(k–4) is HIGH and P2offset(k–5) is HIGH and Te(k–5) is HIGH and Nd(k–6) is LOW and N(k–6) is LOW THEN NOX(k) = 0.04 * N(k–4)–61.83 * P2offset(k–5) + 1.54 * Te(k –5) + 0.48 * Nd(k–6) –0.52 * N(k–6)–251.5
RULE 2 IF (N(k–4) is HIGH and P2offset(k–5) is MED and Te(k–5) is MED and Nd(k–6) is MED and N(k–6) is MED THEN NOX(k) = 0.04 * N(k–4) –53.36 * P2offset(k–5) + 1.2* Te(k–5) + 0.35 * Nd(k–6) –0.37* N(k–6)–207.3
RULE 3 IF (N(k–4) is LOW and P2offset(k–5) is LOW and Te(k–5) is MED and Nd(k–6) is LOW and N(k -6) is LOW THEN NOX(k) = 0.02 * N(k–4)–38.02 * P2offset(k–5) + 1.59 * Te(k –5) + 0.42 * Nd(k–6) –0.54 * N(k–6) + 40.55

The rule set used for this experimental result is shown in Table 3.3.The significance of the vigilance parameter determines the number of clusters, generation of the set of new rules. Thus the adaptation of vigilance parameter remedies the difficulties in choosing apriori values in data clustering**.**

3.5 Learning process

Local learning

Each rule is treated separately. The linear consequent parameters in the weighted least square are used as the normalized membership function values in a fuzzy set. Gaussian radial basis function is good at characterizing local properties. It is essential to trigger higher flexibility for adjoining rules when new operating conditions occur. The use of vigilance parameter steers a tradeoff between plasticity and stability dilemma during the learning process.

Global learning

Neural networks with sigmoidal function are good at characterizing global properties of the learning process. The sigmoidal threshold that is used for the microarray gene datasets is the vigilance threshold.

3.6 Adaptation of vigilance parameter

The adaptive vigilance parameter is used for updation of the cluster surface with each new incoming data point or for the generation of new cluster. Already generated clusters are moved in local areas bounded by the vigilance parameter. A cluster is never initialized far away from the middle of a new upcoming data cloud.

Calculating the Surface of Clusters

If no new cluster needs to be set, the cluster center and the surface of the cluster are updated accordingly. In order to project fuzzy sets form a cluster, the range of influence of the cluster should be available. A good estimation of this range is given by the variance of the data belonging to a nearest cluster. The calculation of this range in incremental mode is accomplished by the recursive variance.

where $\Delta c_{win,j}$ is the distance of the old prototype **Fig**

$$K_{win}\,\sigma^{2}_{winj}(new) = (K_{win}-1)\,\sigma^{2}_{winj}(old) + k_i \Delta c^{2}_{winj} + (c_{winj}(new)-x_j)^2$$

Fig 3.5: Calculating the Surface of Clusters.

$c_{win,j}$(old) to the new prototype $c_{win,j}$(new) of the cluster nearest to the current point x in the jth dimension, and k_{win} is the number of data points lying nearest to cluster $c_{win,}$ can be updated through counting.

New Strategy for Selecting the Winning Cluster

The distance of the new incoming point is calculated incrementally to the surface of the multidimensional ellipsoid spanned by a cluster surface to the nearest cluster center and comparing it with a threshold ρ. It results in a precarious value in the case of wide data clouds: a new data point can lie near or even inside the range of influence, but be away from the center.

Incremental Update of Cluster Centers

The cluster center is updated using the formula in the fig 3.6 where c_{win} is the center in the (p+1) dimensional input/output space nearest to the current data point x and R_{win} the learning gain. The choice of R_{win} plays a central role, as it steers the degree of shifting the centers and results in the convergence of the incremental learning in the least–square.

$$C_{win}\ (new) = C_{win}\ (old) + R_{win}(x - C_{win}(old))$$

$$R_{win} = (init\_gain/k_{win})$$

Fig. 3: Formula for incremental update of cluster center

3.7 Fine tuning process

The optimal selection of vigilance parameter must be tuned in such a way that it must not under fit or over fit the model.

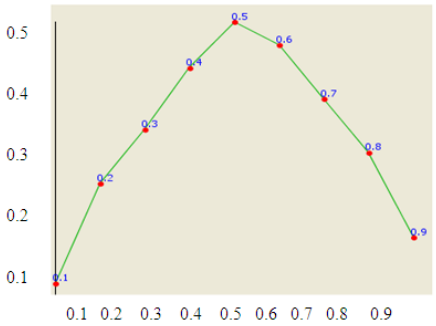

Fig 3.7, the x–axis denotes vigilance parameter and y–axis denotes performance.

While tuning the vigilance parameter it was noted that a value between 0.1 and 0.3 influenced the quality in a significant way. The value between 0.7 and 0.9 tends to over fit the system. A value bouncing between 0.4 and 0.6 influenced the quality of the test data very little hence the vigilance parameter was tuned to an optimal choice of 0.5.

The system may be over trained due to the inappropriate tuning of the vigilance parameter. Over training the network will make it memorize the individual input-

output training pairs rather than settling in the mapping for all cases. A gain term can also be added such that the learning process in training the data set results in unsupervised learning of the network.

Unsupervised clustering of genes and experimental conditions in microarray data can potentially reveal genes that participate in cellular mechanisms that are involved in various diseases. The main underlying assumption is that we can simultaneously obtain better tumor clusters and gene clusters by correlating genes averaged over different samples of the same tumors with the value of the vigilance parameter. Likewise, the correlation of two tumors is more apparent when averaged over sets of genes of similar expression profiles. In situations where the number of tumor types (the number of clusters of experimental conditions) happens to be equal to the number of typical gene profiles (the number of gene clusters), the biclustering algorithm is related to the modified normalized cuts objective function.

4. Performance Analysis

This chapter deals with the performance analysis of this work. The performance analysis is done with respect to two factors viz., misclassification rate and vigilance parameter. The performance of the vigilance parameter in Vector Quantization (VQ) is compared with that of the performance of vigilance parameter in extended incremental version of Vector Quantization (VQ-INC-EXT). This results in the generation of performance analysis graph.

4.1 Misclassification Rate

The misclassification rate is evaluated using the K–Fold Cross-validation technique. The original sample is randomly partitioned into K subsamples. Of the K subsamples, a single subsample is retained as the validation data for testing the system, the remaining K-1 subsamples are used as training data. The Cross validation process is repeated K-times with each of the K subsamples used exactly on the validation data.

The K results from the folds are then averaged to produce a single estimation. The value that is mostly given for the K is ten. Hence this K-Fold Cross-validation technique is otherwise known as 10-Fold Cross-validation technique. ALL and AML datasets are taken as the K subsamples. The raw data in many cancer gene-expression datasets is taken as a validation data for testing the system while the remaining data are partitioned into clusters. The Euclidean distance from the validation data to the winning cluster is calculated. This process is repeated for the remaining samples and all the distances obtained are averaged to produce a single value.

4.2 Vigilance Parameter

Vigilance parameter ρ is used for update of cluster that contain either new incoming data point or generation of new cluster. It has considerable influence in the classification. The higher is the vigilance parameters, the more accurate is the

classification. It also controls the number and size of clusters. In the neural network, it is related to a distance threshold or cluster diameter.

4.3 Comparison of VQ and VQ-INC-EXT

The performance of VQ-INC-EXT is higher when compared with the performance of conventional vector quantization. A graph for misclassification is plotted against the vigilance parameter. In conventional vector quantization the distance is calculated from the new data point to the center of the existing cluster, whereas in the VQ-INC-EXT the distance is calculated from the data point to the surface of the existing cluster. The performance analysis graph is generated for conventional Vector Quantization (VQ) and VQ–INC-EXT.

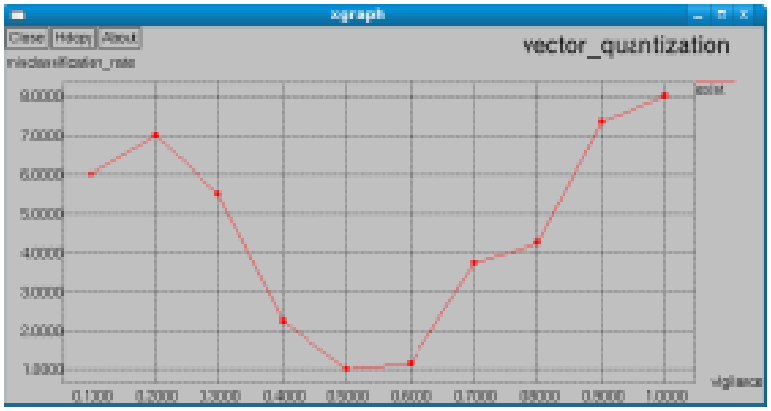

Fig. 4.1: Performance analysis graph for Vector Quantization.

In the fig 4.1 the performance analysis graph for Vector quantization is depicted. The performance of the Vector Quantization is inversely propotional to the rate of misclassification. Hence the performance of the Vector Quantization decreases as the rate of misclassification increases; thus decreasing the quality of the system. The vigilance parameter is made adaptive. It prevents the incorrect cluster partition because of prior setting of the vigilance parameter. This causes a reduction in the misclassification rate of the validation data.

The fig 4.2 shows the performance analysis graph for Extended Incremental version of Vector Quantization (VQ-INC-EXT). The rate of misclassification shown in the performance anlysis graph for VQ-INC-EXT was comparitively lower than the rate of misclassification shown in conventional Vector Quantization. Hence the quality of the system has been increased in the Extended Incremental version of Vector Quantization where the microarray gene datasets show quite satisfactory accurate results in case of new tumor conditions.

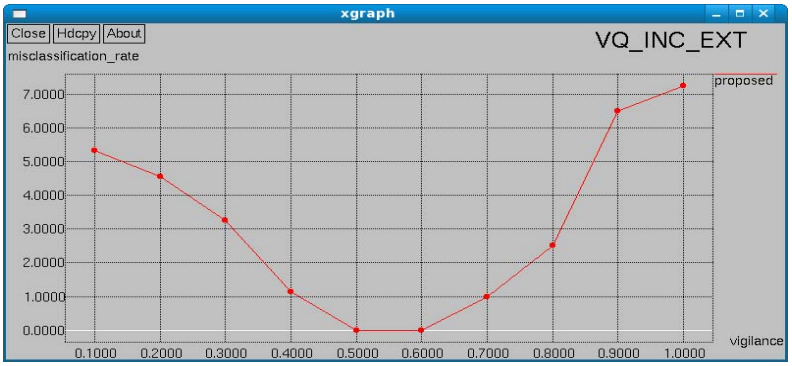

Fig. 4.2: Performance analysis graph VQ-INC-EXT.

Thus, the goal has been accomplished by using VQ-INC-EXT wherein the system can automatically identify the cluster for a new microarray gene expression datasets and its subtypes in online mode and keep high qualitative system up-to-date with newly recorded updated datasets.

5. Conclusion and Future Work

The adaptive vigilance parameter for the clustering part prevents an incorrect cluster partition due to an inappropriate apriori setting of ρ. The system can automatically identify the cluster for a new ALL and AML datasets in online mode and keep high qualitative system up-to-date with newly recorded data. The overall output is obtained via weighted average in Takagi-Sugeno model; thus avoids the time consuming process of defuzzification required in Mamdani model. The selection of highly fit strings helps in obtaining a better, gradual increasing of improving solutions till a desired optimal/sub optimal solution is obtained. The future work may include the following aspects.

i. A forgetting term can be used in the learning process to overcome the limitation of the Fuzzy Inference Engine reaching a saturation point.
ii. Exponential decay of learning rate parameter 'η', guarantees against possibility of Meta stable state.
iii. Identification of overtraining of the network by terminating training once a performance plateau has been reached.
iv. We can use a recursive sample of data for the Leukemia datasets to evaluate the calculation of the potential and rule evolution and replacement strategy based on the updated potentials.
v. It has been shown that removal of irrelevant genes that introduce noise can further improve accuracy in identification of clustering.

vi. If partitioning in the gene dimension is sharper than partitioning in the condition dimension or vice versa, we can organize the conditions or genes of the blurrier dimension contiguously. Such arrangements perhaps give one a sense of the progression of disease states or relevance of a gene to a particular disease.

References

[1] S.Anandhavalli, S.K.Srivatsa, "Improved Takagi Sugeno Fuzzy Modeling with FLEXFIS overlapping Clustering Approach for efficient classification under conflict of interest", International Review on Modeling and Simulation (IREMOS), ,Vol.6, N. 2, April 2013

[2] S.Anandhavalli, S.K.Srivatsa, "Evaluating the quality of test data under the influence of vigilance parameter in FLEXFIS", International Conference on Software Engineering and Mobile Application Modeling and Development, Dec 2012, p 1–15

[3] Chia-Feng and Yu-wei Tsao,"A Self-Evolving Interval Type-2 Fuzzy Neural Network With Online Structure and Parameter Learning", IEEE Trans. Fuzzy Syst., vol. 16, no. 6, pp. 1411-1424,Dec. 2008.

[4] Edwin David Lughofer, "FLEXFIS: A Robust Incremental Learning Approach For Evolving Takagi-Sugeno Fuzzy Models" IEEE Trans. Fuzzy Syst., vol. 16, no. 6, pp 1393–1410, Dec. 2008.

[5] Edwin David Lughofer, "Extension of Vector Quantization for incremental clustering" IEEE Trans. Fuzzy Syst., vol. 14, no. 6, Oct. 2006.

[6] Juan R.Rabunal and Julian Dorado, "Artificial Neural Networking Real Life Application", Idea Group Publishing, Second Edition, 2006,

[7] S. N. Sivanandam, Sumathi , Deepa,"Introduction to Neural Networks Using Matlab 6.0", Tata Mcgraw Hill, 2006

[8] Simon Haykin, "Neural Networks A Comprehensive Foundation", Pearson Education, Second Edition. (2005),

[9] P. Angelov and D. Filev, "Simpl_eTS: A simplified method for learning evolving Takagi–Sugeno fuzzy models," Proc. FUZZ-IEEE 2005, Reno,NV, pp. 1068–1073.

[10] J.S.R.Jang, C.T.Sun, E.Mizutani, "Neuro-Fuzzy and Soft Computing–A Computational approach to learning and Machine Intelligence", Pearson Education, First Edition. 2004,

[11] J.Casillas, O. Cordon, F. Herrera, L. Magdalena, "Interpretability Issues in Fuzzy Modeling", Berlin, Germany: Springer-Verlag, 2003.

[12] Yuval Kluger,1,2 Ronen Basri,3 Joseph T. Chang,4 and Mark Gerstein,"Spectral Biclustering of Microarray Data: Coclustering Genes and Conditions", Genome Research, 2003, Cold Spring Harbor Laboratory Press, pp 703–716

[13] B.Yegananarayana, "Artificial Neural Networks", Prentice-Hall India, Fifth Edition. (2001),
[14] Edwin David Lughofer **"**Evolving Fuzzy Systems–Methodologies, Advanced Concepts and Applications", First Edition, Springer.
[15] R. Babuska,"Fuzzy Modeling for Control", Boston, MA: luwer, 1998.