

# Computational Intelligence Techniques for Acute Leukemia Gene Expression Data Classification

V.P. Plagianakos, D.K. Tasoulis, and M.N. Vrahatis

Department of Mathematics, University of Patras, GR-26110 Patras, Greece,

University of Patras Artificial Intelligence Research Center (UPAIRC),

University of Patras, GR-26110 Patras, Greece

E-mail: {vpp,dtas,vrahatis}@math.upatras.gr

**Abstract**—Recent advances in microarray technologies have allowed scientists to discover and monitor the mRNA transcript levels of thousands of genes in a single experiment. The data obtained from microarray studies present a challenge to data analysis. In this paper, we design an expression-based classification method for acute leukemia. Different dimension reduction techniques are considered to tackle the very high dimensionality of this kind of data. Subsequently, the classification system employs Artificial Neural Networks. The comparative results reported, indicate that high classification rates are possible and moreover that subsets of features that contribute significantly to the success of the neural classifiers can be identified.

## I. INTRODUCTION

In any living cell that undergoes a biological process, different subsets of its genes are expressed. A cell's proper function is crucially affected by the gene expression at a given stage and their relative abundance. To understand biological processes one has to measure *gene expression levels* in different developmental phases, different body tissues, different clinical conditions and different organisms. This kind of information can aid in the characterization of gene function, the determination of experimental treatment effects, and the understanding of other molecular biological processes [17].

Compared to the traditional approach to genomic research, which has been to examine and collect data for a single gene locally, DNA microarray technologies have rendered possible to monitor the expression pattern for thousands of genes simultaneously. Unfortunately, the original gene expression data are contaminated with noise, missing values and systematic variations due to the experimental procedure. Several methodologies can be employed to alleviate these problems, such as Singular Value Decomposition based methods, weighted  $k$ -nearest neighbors, row averages, replication of the experiments to model the noise, and/or normalization, which is the process of identifying and removing systematic sources of variation. Finally, gene expression data are represented by a real-valued expression matrix  $X$ , the rows of which are vectors forming the expression patterns of genes, while the columns represent samples from various conditions. Each cell,  $x_{ij}$ , is the measured expression level of gene  $i$  in sample  $j$ .

The dataset studied here is the well-known and publicly available acute leukemia dataset provided by the center of genome research of the Whitehead Institute [13]. It is a well characterized dataset already used in numerous studies. For

example, in [10] a min-max cut hierarchical clustering method is presented that attempts to produce clusters that are quite close to human expert labeling. Moreover they employ the  $F$ -statistic test and the Principal Component Analysis (PCA) technique for the gene selection. Their approach exhibits classification rates close to the ones presented in this paper.

Discovering the patterns hidden in the gene expression microarray data and subsequently using them to classify the various conditions is a tremendous opportunity and challenge for functional genomics and proteomics [17]. An interesting approach to address this task is to utilize computational intelligence techniques, such as clustering and neural networks. Unfortunately, employing neural networks (or any other classifier) directly to classify the samples is almost impossible due to the *curse of dimensionality* (limited number of samples and very high feature dimensionality). To this end, one has to preprocess the expression matrix  $X$  using a dimension reduction technique [32], or to find a subset of the genes that correctly characterizes the samples.

Clustering can be defined as the process of "grouping a collection of objects into subsets or clusters, such that those within one cluster are more closely related to one than objects assigned to different clusters" [15]. Clustering is applied in various fields including data mining [12], statistical data analysis and social sciences [3], compression and vector quantization [23], global optimization [6], [30], image analysis, and others. Recently clustering techniques have been applied to gene expression data [5], [11], [24], [28] and have proved useful for identifying biologically relevant groupings of genes and samples. Cluster analysis is the key step in understanding how the activity of genes varies during biological processes and is affected by disease states and cellular environments. In particular clustering can be used either to identify sets of genes according to their expression in a set of samples [11], [33], or to cluster samples into homogeneous groups that may correspond to particular macroscopic phenotypes [13]. The latter is in general more difficult, but is very valuable in clinical practice. In [17], a survey of various clustering methods is performed.

The determination of the number of clusters present in a dataset is an unresolved problem in cluster analysis. For instance, well-known and widely used iterative techniques, such as the  $k$ -means algorithm [14], as well as the Fuzzy

$c$ -means algorithm [8], require from the user to specify the number of clusters present in the data prior to the execution of the algorithm. Algorithms that have the ability to discover the number of clusters present in a dataset fall in the category of *unsupervised* clustering algorithms. In this paper we investigate the application of an unsupervised extension of the recently proposed clustering algorithm  $k$ -windows [31] on gene expression microarray data. We have already compared our approach against other well-known clustering algorithms with satisfactory results [26], [27].

In the present work, artificial neural networks are used to classify gene expression data. To tackle the high dimensionality of the data we implement and compare dimension reduction techniques (PCA), as well as supervised (FCM) and unsupervised ( $k$ -windows) clustering algorithms. The clustering algorithms are used to select the most informative features in the dataset, while the PCA technique transforms the original dataset to a lower dimensional projection. The training of the neural networks is performed using an on-line training algorithm [20], [21]. Its key features is that it can handle non-stationary (time-varying) data, while at the same time, it requires relatively little additional memory and computation to process an additional example.

The rest of the paper is organized as follows: in the next section, we briefly describe the principal component analysis, the unsupervised  $k$ -windows algorithm, the supervised fuzzy  $c$ -means algorithm, and the on-line training algorithm. In Section III results from the application of the clustering algorithms on gene expression microarray data are presented. The paper ends with concluding remarks.

## II. METHODS DESCRIPTION

In this section we describe the PCA technique, two cluster analysis methods for feature selection (the supervised fuzzy  $c$ -means algorithm and the unsupervised  $k$ -windows algorithm), and the on-line neural network training algorithm. From the vast field of neural networks research, we focused on Feedforward Neural Networks (FNNs). FNNs have been widely used in pattern recognition tasks, and are considered to be very powerful classifiers compared to classical algorithms such as the nearest neighbor method. More specifically, FNNs have proved capable of finding a good classifier based on a limited and in general small number of training examples [16]. This capability, also referred to as generalization is of particular interest from a pattern recognition point of view since a large set of parameters is estimated using a relatively small data set. For the training of the FNNs we used an on-line training algorithm [20], [21].

### A. Feature Selection Techniques

An important issue in any classification task is to define those features that significantly contribute to the classification of interest, while at the same time discarding the least significant and/or erroneous ones. This procedure is also referred to as dimension reduction. The problem of high dimensionality is often tackled by user specified subspaces of interest. However,

user-identification of the subspaces is error-prone, especially when no prior domain knowledge is available. Another way to address high dimensionality is to apply a dimensionality reduction method to the dataset. Methods such as principal component analysis [18], optimally transform the original data space into a lower dimensional space by forming dimensions that are linear combinations of given attributes. The new space has the property that distances between points remain approximately the same as before.

Another approach is to employ clustering techniques to perform dimension reduction. Specifically, we employed the unsupervised extension of the recently proposed  $k$ -windows [31] clustering algorithm, as well as the supervised Fuzzy  $c$ -means clustering algorithm [8] (FCM) to identify the features of interest. The clustering algorithms were applied over the entire dataset to identify meaningful clusters of features and to select the most informative ones that will be used as input to the FNN. Feature selection was accomplished by extracting from each cluster one representative feature, based on the Euclidean distance among the feature values and the identified cluster center. The feature with the minimum distance from the cluster center was selected.

It must be noted that the user has to supply the FCM algorithm with an estimate of the number of existing clusters (supervised algorithm). On the other hand, the unsupervised  $k$ -windows algorithm automatically approximates the number of clusters present in the dataset. The main contribution of the paper lies in the use of this sophisticated method for selecting quantitative data derived information and in the selection of a robust training algorithm to construct FNNs capable of high accuracy classification.

1) *Principal Component Analysis*: In general, the Principal Component Analysis (PCA) is a powerful multivariate data analysis method. Its main purpose is to reduce and summarize large and high dimensional datasets by removing redundancies and identifying correlation among a set of measurements or variables. It is a useful statistical technique that has found many applications in different scientific fields such as face recognition, image processing and compression, molecular dynamics, information retrieval, and gene expression analysis. PCA is mainly used in gene expression analysis in order to find an alternative representation of the data using a much smaller number of variables, as well as, to detect characteristic patterns in noisy data of high dimensionality. More specifically, PCA is a way of identifying patterns in data and expressing the data in such a way as to highlight their similarities and differences. Since patterns in high dimensional data can be hard to find PCA is a powerful tool of analysis, especially when the visualization of the data is impossible.

Although PCA may succeed in reducing the dimensionality, the new dimensions can be difficult to interpret. Moreover, to compute the new set of dimensions information from all the original dimensions is required. The selection of a subset of attributes in the context of clustering is studied in [1], [2]. In the context of classification, subset selection has also been studied [18].

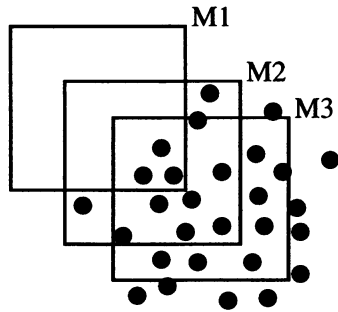


Fig. 1. Sequential movements (solid lines) of the initial window  $M1$  that result to the final window  $M3$ .

2) *Unsupervised  $k$ -windows Clustering Algorithm:* Here we briefly describe the basic concepts of the unsupervised  $k$ -windows algorithm which is applied in this paper. This algorithm extends the  $k$ -windows algorithm [31]. Suppose that we have a set of points in the  $\mathbb{R}^d$  space. Intuitively, the  $k$ -windows algorithm for every cluster present in the dataset tries to place a  $d$ -dimensional window containing patterns that belong to a single cluster.

At first,  $k$  points are randomly selected. The  $k$  initial windows, of size  $a$ , have these points as centers. Subsequently, the patterns that lie within each window are identified. Next, the mean of the patterns that lie within each window is calculated and the window is moved in such a way as its center coincides with the previously computed mean value. The last two steps are repeatedly executed as long as the increase in the number of patterns included in the window that results from this motion satisfies a stopping criterion. The stopping criterion is determined by a variability threshold  $\theta_v$ , which corresponds to the least acceptable change in the center of a window. This process is illustrated in Figure 1.

Once the movement is terminated, the windows are enlarged to capture as many patterns as possible from the cluster. Enlargement takes place at each dimension separately. The windows are enlarged by  $\theta_e/l$  percent at each dimension, where  $\theta_e$  is user defined, and  $l$  stands for the number of previous successful enlargements. After the enlargement in one dimension is performed, the window is moved, as described above. Once movement terminates, the proportional increase in the number of patterns included in the window is calculated. If this proportion does not exceed the user-defined coverage threshold,  $\theta_c$ , the enlargement and movement steps are rejected and the position and size of the window is unchanged.

An example of this process is provided in Figure 2. In this figure the window is initially enlarged horizontally ( $E1$ ). This enlargement is rejected since it does not produce an increase in the number of the included patterns. Next the window is enlarged vertically. This enlargement is accepted and the new window to consider is  $E2$ . The next enlargement reconsidered in the horizontal direction is again rejected.

To automatically approximate the number of clusters present in a dataset a “sufficiently” large number of initial windows is employed and at the final stage of the algorithm overlapping

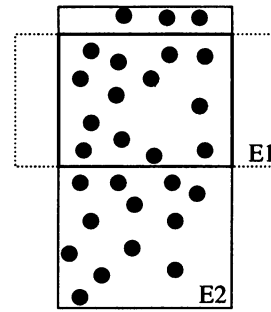


Fig. 2. The enlargement process. Enlargement in the horizontal dimension ( $E1$ ) is rejected, while in the vertical dimension is accepted. After subsequent movements and enlargements the window becomes  $E2$ . Further enlargement in the horizontal dimension is again rejected.

windows that capture patterns belonging to the same cluster are considered for merging. The windowing technique of the  $k$ -windows algorithm allows for a large number of initial windows to be examined without a significant overhead in time complexity. Once all the processes of movement and enlargement for all windows are terminated, all overlapping windows are considered for merging. The merge operation is guided by a merge threshold  $\theta_m$ . Having identified two overlapping windows, the number of patterns that lie in their intersection is calculated. Next the proportion of this number to the total patterns included in each window is calculated. If the mean of these two proportions exceeds  $\theta_m$ , then the two windows are merged (since they considered to belong to a single cluster).

This operation is illustrated in Figure 3; the extent of overlapping between windows  $W_1$  and  $W_2$  exceeds the threshold criterion and the algorithm considers both to belong to a single cluster, unlike windows  $W_3$  and  $W_4$ , which capture two different clusters.

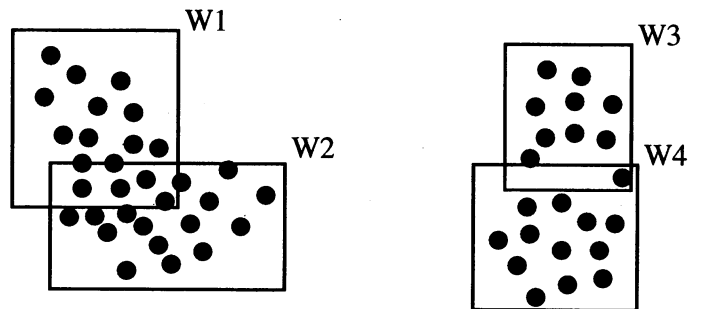


Fig. 3. The merging procedure.  $W_1$  and  $W_2$  have many points in common thus they are considered to belong to the same cluster. On the other, hand  $W_3$  and  $W_4$  capture two different clusters and no merging occurs.

The output of the algorithm is a number of sets that define the final clusters discovered in the original dataset. The computationally demanding step of the  $k$ -windows clustering algorithm is the determination of the points that lie in a specific window. This is the well studied *orthogonal range search* problem [22]. Numerous Computational Geometry techniques

have been proposed [4], [7], [9], [22] to address this problem. These techniques employ a preprocessing stage at which they construct a data structure storing the patterns.

3) *The Fuzzy c-means Clustering Algorithm*: The Fuzzy c-Means (FCM) algorithm [8], considers each cluster as a fuzzy set. It firstly initializes a number of  $c$  prototype vectors (centroids)  $p^j$  over the dataset. The centroids represent the center of the clusters. Next it computes a degree of membership for every data vector  $x^i$  at each cluster using the membership function:

$$\mu_j(x^i) = \left( \sum_{l=1}^c \left( \frac{\|x^i - p^j\|}{\|x^i - p^l\|} \right)^{1/r-1} \right)^{-1},$$

which takes values in the interval  $[0, 1]$ , where  $r \in (1, \infty)$  determines the fuzziness of the partition. If  $r$  tends to  $1_+$ , then the resulting partition asymptotically approaches a crisp partition. On the other hand, if  $r$  tends to infinity, the partition becomes a maximally fuzzy partition. Finally, the  $c$  prototypes are updated using the following equation:

$$p^j = \frac{\sum_{i=1}^n [m_j(x^i)]^r x^i}{\sum_{i=1}^n [m_j(x^i)]^r}.$$

This procedure is iteratively repeated until the measure of the distortion:

$$d = \sum_{j=1}^c \sum_{i=1}^n [m_j(x^i)]^r \|x^i - p^j\|^2,$$

changes less than a user defined threshold.

### B. The Online Neural Network Training Algorithm

Despite the abundance of methods for learning from examples, there are only a few that can be used effectively for on-line learning. For example, the classic batch training algorithms cannot straightforwardly handle non-stationary data. Even when some of them are used in on-line training there exists the problem of “catastrophic interference”, in which training on new examples interferes excessively with previously learned examples, leading to saturation and slow convergence [25].

Methods suited to on-line learning are those that can handle non-stationary (time-varying) data, while at the same time, require relatively little additional memory and computation in order to process one additional example. The following method [20], [21] belongs to this class of methods, and can be used in on-line neural networks training. A high level description of the algorithm is given in Algorithm 1.

In the algorithm model  $\eta$  is the learning rate,  $K$  is the meta-learning rate and  $\langle \cdot, \cdot \rangle$  stands for the usual inner product in  $\mathbb{R}^n$ . As the termination condition the classification error, or an upper limit to the error function evaluations can be used. The key features of this method are the low storage requirements and the inexpensive computations. Moreover, in order to calculate the learning rate for the next iteration, it uses information from the current, as well as, the previous iteration. This seems to provide some kind of stabilization

THE TRAINING ALGORITHM	
0:	Initialize the weights $w^0$ , $\eta^0$ , and $K$ .
1:	<b>Repeat</b>
2:	Set $k = k + 1$
3:	Randomly choose a pattern from the training set.
4:	Using this pattern, calculate the error, $E(w^k)$ and then the gradient, $\nabla E(w^k)$ .
5:	Calculate the new weights using: $w^{k+1} = w^k - \eta^k \nabla E(w^k)$
6:	Calculate the new learning rate using: $\eta^{k+1} = \eta^k + K \langle \nabla E(w^{k-1}), \nabla E(w^k) \rangle$
7:	<b>Until</b> the termination condition is met.
8:	<b>Return</b> the final weights $w^{k+1}$ .

Algorithm 1: The Online Training Algorithm in Pseudocode.

in the calculated values of the learning rate, and previous experiments show that it helps the method to exhibit fast convergence and high success rate.

## III. EXPERIMENTAL RESULTS

The dataset presently considered contains mRNA expression profiles from 72 leukemia patients and has been well characterized [13] aiming at the development of an expression-based classification method for acute leukemia. Each sample is measured over 7129 genes. The first 38 samples were used for the training process (train set), while the remaining 34 were used to evaluate the networks' performance (test set). The initial 38 samples contain 27 acute myeloid leukemia (ALL) samples and 11 acute lymphoblastic leukemia (AML) samples. The test set contains 20 ALL samples and 14 AML samples. The direct application of neural networks to classify the samples is doomed to fail due to the limited number of samples (38) combined with the very high feature dimensionality (7129).

Generally, in a typical biological system, it is often not known how many genes are sufficient to characterize a macroscopic phenotype. In practice, a working mechanistic hypothesis that is testable and largely captures the biological truth seldom involves more than a few dozens of genes, and knowing the identity of these relevant genes is very important [34]. Golub *et al.* in [13] applied the Self Organizing Map [19] (SOM) clustering algorithm on the training set, selecting 50 highly correlated genes with the ALL-AML class distinction. SOM automatically grouped the 38 samples into two classes with one containing 24 out of the 25 ALL samples and the other contained 10 out of the 13 AML samples. Using the techniques proposed here, better results can be achieved.

Firstly, the PCA technique was applied over the entire train set using all 7129 genes. A common problem when using PCA is that there is no clear answer how many factors should be retained for the new dataset. A rule of thumb is to inspect the *scree plot*, i.e. plot all the eigenvalues in their decreasing order. The plot looks like the side of a hill and “scree” refers to the debris fallen from the top and lying at its base. The scree test suggests to stop analysis at the point the mountain (signal) ends and the debris (error) begins. In our case, the scree plot was not decisive (see Fig. 4); it suggested that the contributions are relatively low after the third or after the

tenth component. The FNNs trained using three components exhibited 80% mean classification success, while the FNNs that used ten components obtained 84.12%. However, when we tried 49 different FNNs trained using the datasets having two to fifty components the best performance was exhibited by the FNNs that used five components (92.84% classification success).

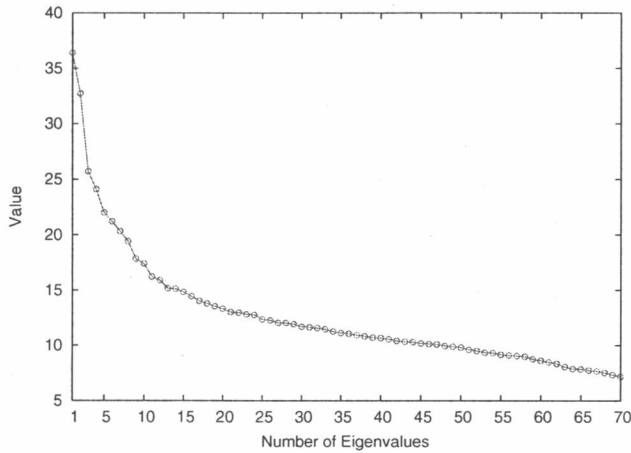


Fig. 4. Plot of the 70 first eigenvalues in decreasing order (scree plot).

Although the FNNs trained using the PCA projection of the dataset can provide high classification accuracy, there is no straightforward interpretation of the new dimensions. Consequently, to compute features for a new patient, information from all the 7129 genes is required. On the other hand, the clustering algorithms identify a subset of genes that significantly contribute to the partition of interest. Thus, only the expression levels of the selected genes are needed for the future operation of the system. Unfortunately, there exist many such subsets and it is difficult for any clustering algorithm to determine the best one.

To this end, we applied the two clustering algorithms over the entire dataset. Regarding the FCM algorithm, we exhibit results for 49 different FNNs using two to fifty features, i.e. the user performed 49 runs instructing the FCM to find two to fifty clusters. Although the classification accuracy of the FNNs was relatively high, it must be noted that the user-defined cluster number has a significant impact on performance. The highest accuracy was obtained by the FNNs trained using 27 representative features, which exhibited a 87.06% mean classification success. In contrast, the FNNs using 24 representative features exhibited the lowest classification accuracy (41.86%).

On the other hand, the unsupervised  $k$ -windows algorithm located automatically ten clusters and the FNNs trained using representative features from those clusters had 87.35% mean classification success. The classification accuracy of the FNNs trained using the transformed datasets provided by the PCA, as well as the subsets of the original dataset provided by the clustering algorithms is illustrated in Fig. 5.

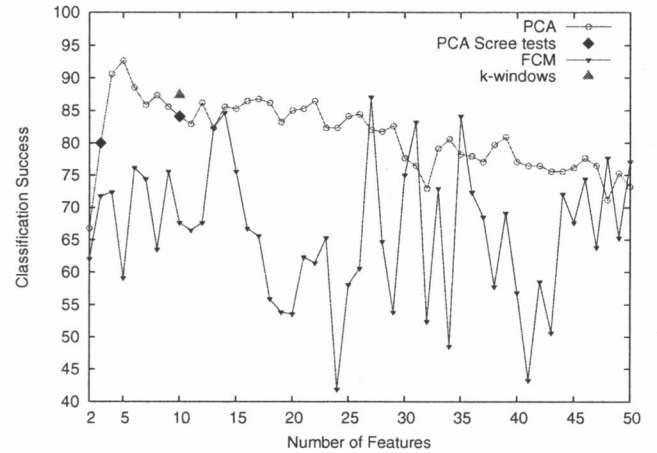


Fig. 5. Classification accuracy of the different methods.

#### IV. CONCLUDING REMARKS

In this paper we compared approaches employing PCA for dimension reduction and clustering techniques in order to find a suitable subset of genes for gene expression microarray data analysis. The dataset used for the experiments was provided by the center of genome research, Whitehead Institute [13]. From the 7129 genes provided, different gene sets were considered. More specifically, cluster analysis groups leukemia samples into clusters based on similar gene expression, while the PCA technique optimally transforms the original data space into a lower dimensional space by forming dimensions that are linear combinations of given attributes. We tested the supervised fuzzy  $c$ -means clustering algorithm and the unsupervised version of the recently proposed  $k$ -windows clustering algorithm, which has proved successful in similar settings [26], [27].

Comparing the different approaches, the best performance was exhibited by the FNNs trained using the the first five components given by the PCA technique (92.65% mean classification success). However, when the scree plot is used to determine the number of PCA components to use, the performance is inferior. The supervised fuzzy  $c$ -means clustering algorithm exhibited the worst results. On the other hand, the unsupervised  $k$ -windows clustering algorithm automatically located ten clusters and the FNNs that used representative features from these clusters exhibited near optimal performance (87.35%), requiring only a single run and no user-defined parameters. We believe that compared with the other approaches the  $k$ -windows algorithm is the approach of choice, when no prior domain knowledge is available. The results of the  $k$ -windows algorithm were overall satisfactory.

Future work will include the implementation of a hybrid algorithm employing PCA and the unsupervised  $k$ -windows algorithm, in an attempt to exploit the advantages each approach has to offer. Preliminary results indicate that FNNs using this idea perform even better and exhibit higher classification success.

## ACKNOWLEDGMENT

The authors thank the European Social Fund (ESF), Operational Program for Educational and Vocational Training II (EPEAEK II) and particularly the Program PYTHAGORAS, for funding this work. This work was supported in part by the “Karatheodoris” research grant awarded by the Research Committee of the University of Patras. We also thank T.R. Golub and his colleagues at MIT for making their AML/ALL dataset available in the public domain.

## REFERENCES

- [1] C.C. Aggarwal, J.L. Wolf, P.S. Yu, C. Procopiuc, and J.S. Park. Fast algorithms for projected clustering. In *Proceedings of the ACM SIGMOD international conference on Management of data*, 61–72, 1999.
- [2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM SIGMOD international conference on Management of data*, 94–105, 1998.
- [3] M.S. Aldenderfer and R.K. Blashfield. *Cluster Analysis*, volume 44 of *Quantitative Applications in the Social Sciences*. SAGE Publications, London, 1984.
- [4] P. Alevizos. An algorithm for orthogonal range search in  $d \geq 3$  dimensions. In *Proceedings of the 14th European Workshop on Computational Geometry*, Barcelona, 1998.
- [5] 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 array. *Proceedings of the National Academy of Sciences USA*, 96(12):6745–6750, 1999.
- [6] R.W. Becker and G.V. Lago. A global optimization algorithm. In *Proceedings of the 8th Allerton Conference on Circuits and Systems Theory*, 3–12, 1970.
- [7] J.L. Bentley and H.A. Maurer. Efficient worst-case data structures for range searching. *Acta Informatica*, 13:1551–68, 1980.
- [8] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, 1981.
- [9] B. Chazelle. Filtering search: A new approach to query-answering. *SIAM Journal on Computing*, 15(3):703–724, 1986.
- [10] C. H. Q. Ding. Analysis of gene expression profiles: class discovery and leaf ordering. In *Sixth annual international conference on Computational biology*, 127–136. ACM Press, 2002.
- [11] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences USA*, 95:14863–14868, 1998.
- [12] U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. *Advances in Knowledge Discovery and Data Mining*. MIT Press, 1996.
- [13] T.R. Golub, D.K. Slomin, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Collier, M.L. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.
- [14] J.A. Hartigan and M.A. Wong. A  $k$ -means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.
- [15] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.
- [16] S. Haykin. *Neural Networks*. New York: Macmillan College Publishing Company, 1999.
- [17] D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 16:1370–1386, 2004.
- [18] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant Features and the Subset Selection Problem. In *Proceedings of the International Conference on Machine Learning*, 121–129, 1994.
- [19] T. Kohonen. *Self-Organized Maps*. Springer Verlag, New York, Berlin, 1997.
- [20] G.D. Magoulas, V.P. Plagianakos, and M.N. Vrahatis. Hybrid Methods Using Evolutionary Algorithms for On-line Training. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN'2001)*. Washington D.C., U.S.A., 2001.
- [21] V.P. Plagianakos, G.D. Magoulas, and M.N. Vrahatis. Global Learning Rate Adaptation in On-line Neural Network Training. In *Proceedings of the Second International ICSC Symposium on Neural Computation (NC'2000)*. Berlin, Germany, 2000.
- [22] F. Preparata and M. Shamos. *Computational Geometry*. Springer Verlag, New York, Berlin, 1985.
- [23] V. Ramasubramanian and K. Paliwal. Fast  $k$ -dimensional tree algorithms for nearest neighbor search with application to vector quantization encoding. *IEEE Transactions on Signal Processing*, 40(3):518–531, 1992.
- [24] R. Shamir and R. Sharan. Click: A clustering algorithm for gene expression analysis. In *8th International Conference on Intelligent Systems for Molecular Biology (ISMB'2000)*. AAAIPress, 2000.
- [25] R.S. Sutton and S.D. Whitehead. Online learning with random representations. In *Proceedings of the Tenth International Conference on Machine Learning*. Morgan Kaufmann, 314–321, 1993.
- [26] D.K. Tasoulis, V.P. Plagianakos, and M.N. Vrahatis. Unsupervised cluster analysis in bioinformatics. In *Fourth European Symposium on “Biomedical Engineering”*, 2004.
- [27] D.K. Tasoulis, V.P. Plagianakos, and M.N. Vrahatis. Unsupervised clustering of bioinformatics data. In *European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems, Eumite*, 47–53, 2004.
- [28] S. Tavazoie, J.D. Hughes, M.J. Campbell, R.J. Cho, and G.M. Church. Systematic determination of genetic network architecture. *Nature Genetics*, volume 22:281–285, 1999.
- [29] J.G. Thomas, J.M. Olson, S.J. Tapscott, and L.P. Zhao. An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles. *Genome Research*, 11:1227–1236, 2001.
- [30] A. Torn and A. Zilinskas. *Global Optimization*. Springer-Verlag, Berlin, 1989.
- [31] M.N. Vrahatis, B. Boutsinas, P. Alevizos, and G. Pavlides. The new  $k$ -windows algorithm for improving the  $k$ -means clustering algorithm. *Journal of Complexity*, 18:375–391, 2002.
- [32] M.E. Wall, A. Rechtsteiner, and L.M. Rocha. Singular value decomposition and principal component analysis. *A Practical Approach to Microarray Data Analysis*, chapter 5, Kluwer, 91–109, 2003.
- [33] X. Wen, S. Fuhrman, G. Michaels, D. Carr, S. Smith, J. Barker, and R. Somogyi. Large-scale temporal gene expression mapping of cns development. *Proceedings of the National Academy of Science USA*, 95:334–339, 1998.
- [34] E.P. Xing and R.M. Karp. Cliff: Clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics Discovery Note*, 1:1–9, 2001.