International Joint Conference on Neural Networks 2004

July 31-August 4, 2005
Hilton Montreal Bonaventure Hotel
Montréal, Québec, Canada

Societies and Sponsors:

INTERNATIONAL NEURAL NETWORK SOCIETY

IEEE

IEEE Computational Intelligence Society

Florida Institute of Technology

Ford Motor Company

Applied Computational Intelligence Laboratory, U
Missouri-Rolla

University of Texas at Arlington

Siemens

Cisco

IJCNN 2005 Conference Program

Special Session Sb: Neural Networks Applications to Bioinformatics

Monday, August 1, 9:30AM-11:30AM, Room: Westmount, Chair: Francesco Masulli and Roberto Tagliaferri

9:30AM Data Visualization Methodologies for Data Mining Systems in Bioinformatics [1469]
Antonino Staiano, Angelo Ciaramella, Giancarlo Raiconi, Roberto Tagliaferri and Giuseppe Longo

9:50AM Random projections for assessing gene expression cluster stability [1554]
Alberto Bertoni and Giorgio Valentini

10:10AM A New Approach to Hierarchical Clustering for the Analysis of Genomic Data [1455]
Francesco Masulli and Stefano Rovetta

10:30AM Inferring Protein-Protein Interactions Using Interaction Network Topologies [1592]
Alberto Paccanaro, Valery Trifonov, Haiyuan Yu and Mark Gerstein

10:50AM Predicting sugar regulation in Arabidopsis thaliana using kernel learning methods [1472]
Kamel Saadi, Kee-Khoon Lee, Gavin Cawley and Michael Bevan

11:10AM Feedback Linearization Using Neural Networks Applied to Advanced Pharmacodynamic and Pharmacogenomic Systems [1628]
Alexandru Floares

A New Approach to Hierarchical Clustering for the Analysis of Genomic Data

Francesco Masulli
Dept of Computer Science
University of Pisa
Largo B. Pontecorvo, 3 I-56125 Pisa, Italy
masulli@di.unipi.it

Stefano Rovetta
Department of Computer and Information Sciences
University of Genova
Via Dodecaneso 35 I-16146 Genova, Italy
rovetta@disi.unige.it

***Abstract*— Clustering algorithms in biomedical disciplines are usually selected between two main families, k-Means and Agglomerative Hierarchical Clustering. These methods are well studied and well established. However, both categories have some drawbacks related to data dimensionality (for partitional algorithms) and to the bottom-up structure (for hierarchical algorithms). To overcome these limitations, we present a hierarchical clustering algorithm based on a completely different principle, which is the analysis of shared *farthest neighbors*. The principle of operation and the rationale are illustrated, and experimental results on different data sets are presented.**

I. Introduction

Clustering algorithms in biomedical disciplines are usually selected between two main families. When the number of experimental observations (cardinality) is high and the number of observed variables (dimensionality) is not very large, it is possible to use iterative, partitional algorithms such as k-Means [1]. When data dimensionality is very large, or the number observations is small, then hierarchical agglomerative algorithms [2] are normally used. Popular examples include average linkage [3] methods and Ward's method [4].

All of these methods are well studied and established. In the hierarchical cases, the resulting tree structure can be easily represented in visual form as a dendrogram [5] or color diagram [2]. However, both categories have some drawbacks related to data dimensionality (for partitional algorithms) and to the bottom-up structure (for hierarchical algorithms).

To overcome these limitations, we present a hierarchical clustering algorithm based on a completely different principle, which is the analysis of shared *farthest neighbors*. This approach share some similarities with Jarvis-Patrick clustering [6], which however is based on the analysis of shared *nearest neighbors* and is not a hierarchical method.

II. Limitations of current methods

The clustering methods usually adopted have their own weaknesses, which we are going to point out in this section. In the following section we will propose countermeasures.

The k-Means clustering method is one of the most popular. It is well known that it is prone to local minima; however, when data are sampled in sufficient quantity and the number k of centroids is small, this may not be a problem.

However, with biomedical data, the typical situation is that a single experimental observation is very expensive, therefore many variable are observed at any experiment. This is exactly the situation we have with genomic data, and even more so with microarray experiments. This raises the issue of the curse of dimensionality [7][8], that is, the need for exponentially many data points as a function of space dimensionality.

The k-Means algorithm (as well as any one of its many variants) searches for regions where data are especially dense. However we expect that, when clustering experiments, the cardinality of the data sets available is not only small with respect to the size of the data space (dimension equal to the number of variables), which would lead to insufficient sampling of the space: it is usually even less than the number of variables. This means that the data span only a subspace within the data space. In these conditions, it is not even easy to define the concept of (hyper)volumetric density, let alone estimating it. Therefore k-Means is typically adequate for clustering variables across experiments, rather than clustering experiments.

There have been many efforts in solving the dimensionality problem. We refer the reader for instance to [9], in which the problem is tackled by seeking clusters on subspaces of the data space (projection clustering), and to the literature cited therein for other examples.

Another drawback, shared by both types of techniques, is related to the number of clusters. The "k" in k-Means, number of expected clusters, should be known in advance, or estimated either by prior knowledge or by a-posteriori validation. These approaches are implemented in many variants, such as ISODATA [10].

The standard hierarchical approach, on the other hand, makes no attempt at defining proper clusters, leaving up to subsequent analysis to split the hierarchy into clusters (at appropriate levels). The problem is related to the binary procedure of agglomeration, whereby at each stage only two objects (clusters or data objects) are merged. In this sense, we could even say that this it is not a true clustering technique, since it does provide an organization of data, but with no attempt at identifying clusters. The method retains its usefulness in its ability to visualize a structure in the form of a taxonomy, which makes it interesting in a number of applications.

A derivative problem, which is produced by clusters being obtained only a-posteriori in hierarchical techniques, is that the taxonomy obtained is not very robust with respect to noise. In the presence of perturbations, different taxonomic trees can be obtained even if the perturbations are small. Usually this problem is tackled with resampling approaches (bootstrap), but this is again an a-posteriori remedy.

A further problem is again related to space dimensionality. Defining clusters on the basis of distance requires that distances can be estimated. However there are results [11] stating that, when space dimensionality is high or even moderate (as low as 10-15), the distance of a point to its farthest neighbor and to its nearest neighbor tend to become equal. This causes the actual evaluation of distances, and the concept of "nearest neighbor" itself, to become less and less meaningful with growing dimensionality.

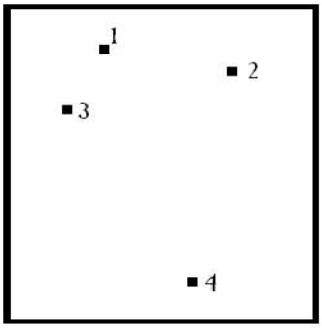
We can also add to the list the minor inconvenience, for agglomerative methods, of not being able to produce a partial (rough) result, to be refined only if needed. In the data mining jargon, algorithms with this property are called "anytime" algorithms. In our case this may not be a significant factor from the computational point of view, but it depends on the application.

III. Addressing the limitations

We outline now the remedies which we propose to overcome the limitations above. To avoid the problems with the number of clusters, an algorithm should be hierarchical, but at the same time it should allow for more than two objects at any level in the hierarchy. The procedure should be divisive rather than agglomerative, which produces an algorithm that we can use up to a desired level of detail without being constrained to proceed to the level of single data objects. In this way, the criterion used to divide each cluster into (possibly more than two) subclusters provides an indication of the "appropriate" number of clusters for that level in the hierarchy, although assessing that this number is the true number of natural clusters would typically require further analysis. For instance, our choice is an indirect approach whereby class labels in supervised problems are used to validate clustering results (see Section VI).

To tackle the dimensionality problem, a typical countermeasure found in traditional statistics is moving from the analysis of values (in our case, distances) to the analysis of their *ranks*. Rank is the position of a given value in the ordered list of all values. This technique is adopted when using actual values is either difficult or inadequate. The approach is followed for instance by Spearman with his rank-correlation index r_s [12] or by Kendall with his correlation index τ and coefficient of concordance W [13].

Regarding noise robustness, it is certainly possible to apply some technique to filter out noisy samples and outliers. This however requires prior knowledge on the statistics of data, so that the definition of noise and outliers allows labeling as such those points which do not reflect this statistics. This approach is not very attractive from the viewpoint of the present paper. We try to avoid the assumption of a given distribution or cluster shape, both because this would limit the generality of the approach we are proposing, and because very small samples are not statistically significant. Besides, filtering out what we have labeled as noise may throw away relevant information which might be important in an exploratory data analysis step, and reducing an already scarce data set is not advisable anyway.



Data points	1	2	3	4
I Neighbor	3	1	1	3
II Neighbor	2	3	2	2
III Neighbor	4	4	4	1

Fig. 1. An example data set to illustrate the "Points in perspective" principle. For each point the table lists the distance ranks of all other points.

The following section will present a principle of operation based on these considerations and the resulting clustering algorithm.

IV. The "Points in Perspective" Principle

We propose to adopt the following principle of operation: *Two points should be considered similar if they share the same farthest point among all remaining data.* We term this the "Points in Perspective" Principle, since the points are examined not with reference to their neighborhood (locally), but with reference to far-away points in the data set, therefore in perspective.

Note that usually similarity is assessed on the basis of the nearest neighbor. For instance, k-Means clustering is done by associating each data point to the closest cluster centroid or prototype. However, we do not want to resort to centroids to define clusters, since this would limit the procedure to metric data only, and since this would require estimation of centroids (which we have seen to be an ill-posed problem when the data set is of lower cardinality than dimensionality). This leaves open the option of the Jarvis-Patrick [6] approach, termed "shared nearest neighbors" (SNN). Points are considered similar if they share the nearest neighbor, or a list of a given number of nearest neighbors.

However, the SNN produces the following odd result. The higher the rank of neighbors, the larger their "agglomerative" significance. Two points which are very close to each other and distant to other data points should be considered as a good cluster. But since the (first) nearest neighbor of either

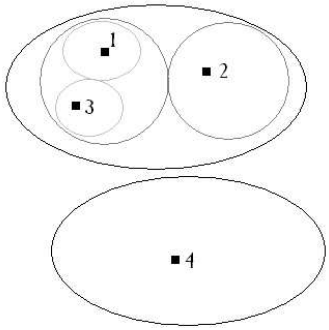


Fig. 2. The example data set clustered according to the proposed method.

point is the other point, the first nearest neighbor is *always different*. This of course is not a major drawback (SNN simply counts k neighbors and groups objects with at least k_t shared neighbors), but it offers some evidence that the principle itself may be only partially justified.

Moreover, the SNN approach can be unreliable with very sparse data, where clusters may be sampled by only one or two objects. This poses the issue of selecting the clustering threshold k_t.

As a last remark, we recall that we are seeking a hierarchical method, and SNN provides only partitional clustering, although in the original presentation the authors suggest repeated applications of the method to obtain tree-structured clusters.

The example shown in Figure 1 clarifies these remarks.

Clustering according to the proposed "Points in Perspective" principle of operation is done according to the following very simple procedure: First, all points are labeled. We compute the distance of each point to all others, and for each point we identify the farthest neighbor. We define clusters at the first level by aggregating all points sharing a common farthest neighbor label.

We should point out that, although we are discussing the method in terms of distance, it is applicable to more general dissimilarity definitions than proper distance.

Then, within each cluster, the second farthest neighbor can be considered exactly in the same way. This produces a second level clustering within each cluster of the first level.

The procedure is recursively repeated until no further differentiation is found (all points within a level $l-1$ cluster share the same l-th farthest neighbor), or until a predefined maximum level is reached.

We term this algorithm *Shared Farthest Neighbor* clustering (SFN). The example shown in Figure 2 illustrated the result of applying the SFN procedure to the data of Figure 1.

Here a proposed implementation of the SFN algorithm is sketched. The algorithm starts computing the distance matrix D (the matrix of distance between each point i and each point j). This is not a symmetric matrix in the case of more general dissimilarity measures, but the method does not require symmetry and remains applicable in more general cases. Note also that, if we have a *similarity* measure instead of dissimilarity, we only need to change the direction of the comparison used in the sorting phase. Finally, this is the phase where, if required, we can take care of missing data by adequately defining the measure (imputation seems a less appropriate technique, given the small cardinality which we have set as our starting hypothesis).

Once we have D, which may also be given as the input to the algorithm, we proceed as follows. For each point in the data set (a row of D) the distances to other data points are ordered and the corresponding rank is written in place of the actual distance, obtaining a rank matrix R.

Now each row of this matrix should be "inverted", that is, cell contents should be swapped with the corresponding cell indexes. We obtain an index matrix X listing, for each data point, all point labels in order of distance. This can be done simply looking up ranks in R and writing point labels in the corresponding position of X (that is, $x_{i,r_{ij}} = j$).

Clustering is now performed simply by sorting the rows of matrix X. Conceptually, this is done according to each column, starting from the first (nearest neighbors) up to the last. However, we can decrease the algorithm complexity as a function of data cardinality, and at the same time allow for a partial clustering, i.e. stopping at a given level of the hierarchy. This can be obtained if we start sorting from the farthest neighbor, then partially sort the rows within each individual cluster, and so on.

The following algorithm summarizes the procedure. The method can be implemented in many ways, but this pseudocode reflects the structure of the implementation which is available at the web address http://mlsc.disi.unige.it/C/sfn/.

```
-----------------------------------------
Algorithm SFN
Data structures:
   matrix D (n x n)
   matrix X (n x n)
   matrix R (n x n)
Input:
  training set T (cardinality n)
Compute D = distance matrix for T
Compute R = ranks of distances in D
|  (within each row)
Compute X = index matrix
|  (by swapping cell contents with indexes in R)
for i = n to 1 {
   Sort rows of Y using column i as key
}
Output:
   clusters
|  clusters at hierarchical depth i share the
|  same value in column i of matrix Y
end algorithm
-----------------------------------------
```

V. Properties of the proposed approach

In this section we highlight some features of the approach presented and of the resulting algorithm.

The algorithm implemented according to the above sketch is of the "anytime" type, because it is a *divisive* technique, not an agglomerative one. We can decide to stop it when the hierarchy is partially built, and obtain a usable clustering result. Usually it is advisable to make use of this property, so that the result is more understandable (fewer larger clusters). It also makes little sense to split clusters into extremely small partitions when the data set is already scarce.

With respect to the position of points and to its perturbations, the hierarchy of dichotomies is more stable than in hierarchical agglomerative clustering algorithms. This is because clustering is based on the largest distances, over which the effect of small perturbations is usually negligible, rather than on the smallest.

A cluster is not constrained to be separated in exactly two sub-clusters, and the clustering structure is therefore allowed to fit the natural structure of data (that can be non-dichotomic).

After the distance matrix D has been obtained, the algorithm operation (and computational complexity) is independent on data dimensionality. On the other hand, the dependence on the data cardinality (number of points) is not important, since by design we are in the case of small cardinality. Moreover, distances in the data space are used only for computing ranks and not for estimating densities or approximating region geometries. Therefore the algorithm is especially appropriate in those situations where cardinality is low and dimensionality is high. This makes it well suited to the analysis of genomic data, for instance with DNA microarrays. In general, many bio-medical data analysis problems fall within this category, and the algorithm can be successfully applied.

VI. Experimental validation

We have validated the SFN algorithm on medical diagnosis and genomic data analysis problems, some of which are publicly available. The data sets include:

1) Lung cancer. Five patients with lung cancer have been analyzed with a DNA microarray technique. These are preliminary results from an on-going study and are not publicly available. Given the very small cardinality, these data have been used to validate the method against the results obtained with hierarchical agglomerative clustering.
2) Pima Indians diabetes [14]. Pima Indians are affected by an endemic form of diabetes, which is found with much higher frequency than in other populations, and have agreed to be the subject of a study. The data collected have been put in the public access on the UCI repository of machine learning databases [15].
3) Wisconsin diagnostic breast cancer [16]. Samples of breast mass are microscopically analyzed. The data are obtained by digitizing an image from each sample. Features describe the cell nuclei present in the image. These data are from the UCI repository as above.
4) Lyme disease [17][18]. A disease discovered in the relatively recent past. It has initial effects on skin, then it can reach the nervous system, heart, connective tissue (Lyme arthritis). In regions where it is not endemic, the diversity of signs can be confusing even to medical professionals trying to diagnose it, if they are not specifically trained. One of the authors has worked on this data set, which is currently not publicly available.
5) Molecular classification of leukemia [19]. DNA microarray are used to characterize two forms of leukemia at the molecular level, and within one of the two forms to separate two further sub-classes which are not distinguishable at the morphologic or serologic level, but have dramatically different prognoses. There are a training set and a test set, both available from the web address `http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi`.
6) Splice junction sequences [20]. Splice junction sites are point in the genome where introns (non-coding sequences) and exons (coding sequences) are joined together. The task is to identify splicing sites. These data have been obtained from the UCI repository as above.

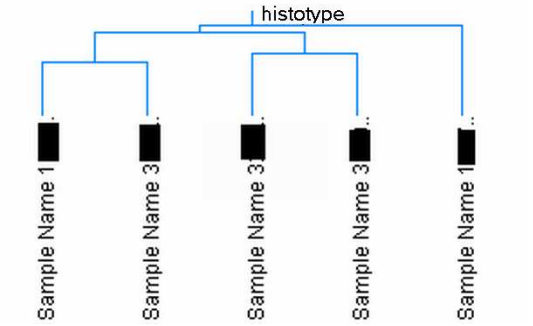


Fig. 3. Dendrogram obtained on Problem 1 by the SFN algorithm and by hierarchical agglomerative clustering.

Please note that full documentation and credits for the publicly accessible databases, as well as references to the relevant literature, should be obtained from the cited sources. The Lung cancer data are **not** the same as the data set with the same name available from the UCI repository.

The first experiment consists in validating the clustering result on problem 1. This is to achieve a first indication that the clusters we get are reasonable. This problem has a very small data cardinality, so the number of possible clusterings is limited and, arguably, there is only one "correct" result.

Figure 3 show the dendrogram obtained with the SFN algorithm and with hierarchical agglomerative clustering. Similarity is defined as the correlation between data points. We obtain the same result in both cases, which is therefore shown only once. The picture is a screenshot from the commercial data analysis package which produced the hierarchical agglomerative dendrogram.

The experimental results reported on Table 1 are obtained on Problems 2–5, all with Euclidean distance.

To evaluate the quality of clustering, we adopt the following approach. The result of clustering is usually assessed on the basis of some external knowledge about how clusters should

TABLE I

EXPERIMENTAL RESULTS

Problem	n	Preprocessing	Error %
2	768	Normalized with respect to average/stdev	12.40%
3	569	Normalized with respect to average/2*stdev	5.60%
4	684	Normalized with respect to average/2*stdev	6.00%
5 (training set)	38	none	0.00%
5 (training+test sets)	72	none	6.90%

be structured. This may imply evaluating separation, density, connectedness, diameter, and so on. However, these are all evaluations of results against a given expectation, which may not translate into good performance when the method is applied to a problem.

The only way to assess the usefulness of a clustering result is indirect validation, whereby clusters are applied to the solution of a problem and the correctness is evaluated against objective external knowledge. For this reason we need labeled data sets, where the external knowledge is the class information provided by labels. The experiments are therefore all performed on supervised problems.

We expect that, if the algorithm finds significant structures in the data, these will be reflected by the distribution of classes. Therefore we operate a "calibration" step for clusters (assigning to each cluster the class label which is most represented among its data points) and compare them to the behavior of *supervised* methods from the literature.

In this way we cannot obtain a direct assessment of the goodness of clusters per se; in exchange, we obtain valuable information about how these clusters map on the natural structure of the problem.

Regarding the evaluation method, we choose not to perform cross-validation or similar procedures, considering that the algorithm is "trained" in a completely unsupervised manner, and calibration already occurs (in a sense) on an external validation data set, which is the set of class labels. Cross-validation or resampling methods, however, can be very useful to assess the stability of the proposed method, by comparing clustering structures in repeated experiments.

The results we achieve are comparable with those obtained by supervised approaches proposed in the literature. This should be a confirmation of the validity of the method. Since clustering is done in a completely unsupervised manner, finding that the cluster structure is reasonably mapped onto the true classes supports the hypothesis that the algorithm is capable of discovering the "true" structure, the one which is inherent in data.

In particular, the results on the Leukemia dataset show that the method compares favorably with the approach by Golub et al. [19]. For instance, performance on the training set of 38 samples is errorless in our case, whereas the original self-organizing map (SOM) approach yielded 4 misclassified samples.

It is not easy to compare the deeper trees obtained by standard agglomerative hierarchical clustering to those obtained with the proposed method, which may be much less deep and still convey significant structure, since they have no constraint on the number of subclusters. In the case of Leukemia data, the tree depth for standard hierarchical clustering is at least 6 (for instance, with the average linkage method we obtain a tree depth of 9). For SFN, splitting stopped at level 4, although only 1 cluster was split up to the fourth level, whereas 12 clusters with no further sub-structure were present at level 1. Calibration itself is not a well-defined process for a binary tree, since the structure of clusters is not related to the depth of the tree, but rather to the linkage value. The tree should therefore be trimmed to a given (arbitrary) linkage value.

We can comment further on the clusters obtained by taking also into account the class labels, which are "ALL" for 27 Acute Lymphoblastic Leukemia patients and "AML" for 11 Acute Myeloid Leukemia. The distribution of cardinality among the clusters at level 1 is as follows:

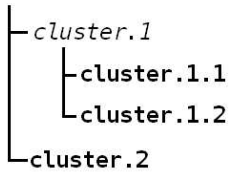
Cardinality	**Clusters**	**Class**
10	1	AML
5	1	ALL
4	1	ALL
2	5	ALL
1	4	ALL

The cluster with further structure had cardinality 5 and contained one data object of class AML. All other AML were in the largest level 1 cluster. All leaf clusters (those which are not further split) are homogeneous with respect to the diagnosis.

This suggests a structure in data whereby AML profiles are better characterized than ALL profiles. This is clearly true when we notice that there are two sub-classes of ALL, which are T-cell ALL and B-cell ALL.

The distribution in general is well represented by a partitional clustering (this is a confirmation of the already good result obtained by Golub *et al.* with the SOM approach), However there is a subset of the data which needs a deeper structure for adequate representation. After the calibration step, we see that this subset contains a sample diagnosed as AML which is correctly separated from the other samples. Cluster structure is again confirmed by the class labels.

Problem 6 is different in that it involves data objects which are not metric vectors, but strings of DNA sequences, 60 bases long and centered around the candidate splicing site. Distance here is defined as the number of mismatches between bases in corresponding positions (only the 40 central bases have been considered). Here the result is very good: Figure 4 illustrates the hierarchy obtained (graphics from a program by the authors). Fixing the maximum level at 2, the structure is very simple, with a cluster further split into two sub-clusters and another cluster without sub-clusters. Yet the resulting



count	pos	neg	class	perc
2495	1593	902	1	63.8%
902	0	902	-1	100.0%
1593	1593	0	1	100.0%
695	0	695	-1	100.0%

Fig. 4. Performance on Problem 6. In the diagram: clusters in slanted font are further split into sub-clusters. In the table: *count* is number of objects in each cluster; *pos* and *neg* indicate splicing and non-splicing sites, resp.; *class* is the majority class; *perc* is the percentage of objects in the majority class.

classification, after performing the calibration step, is errorless, as indicated in the figure.

These data should be compared to results of other methods. Among the results reported in the accompaining documentation to the data set, no supervised method is capable of errorless performance. Comparison with centroid-based clustering methods (k-Means) is not possible, since a proper centroid (barycenter) is not obtainable from non-metric data. It is also difficult to compare the obtained tree to that given by the standard agglomerative hierarchical methods, since, in contrast to Problem 1, here the cardinality is high as an absolute value (although still very low when related to the dimensionality). Trees obtained with these methods may or may not be comparable to the one presented.

VII. Conclusion

The clustering algorithm presented here is based on a novel principle of operation, and as such has properties not found in other more commonly used methods. It is especially designed for the analysis of data sets with high dimensionality and low cardinality, and is therefore well suited to DNA microarray data analysis, as demonstrated by the experiments. However it is more generally applicable in the field of biomedical data analysis, where these conditions are often met.

We have observed that, similarly to the Jarvis-Patrick algorithm, the method presented may suffer from many small or singleton clusters. This happens especially when data cardinality grows. Future developments include criteria for controlling the proliferation of singletons (cluster validity) and applications to outlier detection.

Acknowledgment

Work funded by the Italian National Institute for the Physics of Matter (INFM), the Italian Ministry of Education, University and Research (2004 “Research Projects of Major National Interest”, code 2004062740), and the Biopattern EU Network of Excellence.

References

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review”, *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[2] 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*, vol. 95, no. 25, pp. 14863–14868, 1998.

[3] R. R. Sokal and C. D. Michener, “A statistical method for evaluating systematic relationships”, *University of Kansas Science Bulletin*, vol. 38, pp. 1409–1438, 1958.

[4] J. H. Ward, “Hierarchical grouping to optimize an objective function”, *Journal of American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.

[5] R. R. Sokal and P. H. Sneath, *Principles of Numerical Taxonomy*, Freeman, San Francisco, USA, 1963.

[6] R. A. Jarvis and E. A. Patrick, “Clustering using a similarity measure based on shared near neighbors”, *IEEE Transactions on Computers*, vol. C22, pp. 1025–1034, 1973.

[7] R. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, 1961.

[8] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, New York (USA), 1973.

[9] C. C. Aggarwal and P. S. Yu, “Redefining clustering for high-dimensional applications”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 210–225, March/April 2002.

[10] G. H. Ball and D. J. Hall, “ISODATA, an iterative method of multivariate analysis and pattern classification”, *Behavioral Science*, vol. 12, pp. 153–155, 1967.

[11] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, “When is nearest neighbor meaningful?”, in *7th International Conference on Database Theory Proceedings (ICDT’99)*. 1999, pp. 217–235, Springer-Verlag.

[12] C. Spearman, “‘General intelligence,’ objectively determined and measured”, *American Journal of Psychology*, vol. 15, pp. 201–293, 1904.

[13] M. Kendall and J. D. Gibbons, *Rank Correlation Methods*, Oxford University Press, Oxford (UK), fifth edition, 1990.

[14] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes, “Using the adap learning algorithm to forecast the onset of diabetes mellitus”, in *Proceedings of the Symposium on Computer Applications and Medical Care*. 1988, pp. 261–265, Computer Society Press.

[15] C. L. Blake and C. J. Merz, “UCI repository of machine learning databases”, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[16] W. N. Street, W. H. Wolberg, and O. L. Mangasarian, “Nuclear feature extraction for breast tumor diagnosis”, in *IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology*, 1993, vol. 1905, pp. 861–870.

[17] C. Moneta, G. Parodi, S. Rovetta, and R. Zunino, “Automated diagnosis and disease characterization using neural network analysis”, in *Proceedings of the 1992 IEEE International Conference on Systems, Man and Cybernetics - Chicago, IL, USA*, October 1992, pp. 123–128.

[18] G. Bianchi, L. Buffrini, P. Monteforte, G. Rovetta, S. Rovetta, and R. Zunino, “Neural approaches to the diagnosis and characterization of lyme disease”, in *Proceedings of the 7th IEEE Symposium on Computer-Based Medical Systems, Winston-Salem, NC*, 1994, pp. 194–199.

[19] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, “Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring”, *Science*, vol. 286, no. 5439, pp. 531–537, October 1999.

[20] M. O. Noordewier, G. G. Towell, and J. W. Shavlik, “Training knowledge-based neural networks to recognize genes in dna sequences”, in *Advances in Neural Information Processing Systems III*. 1991, vol. 3, Morgan Kaufmann.