Effectiveness of Different Partition Based Clustering Algorithms for Estimation of Missing Values in Microarray Gene Expression Data

Shilpi Bose[1], Chandra Das[1], Abirlal Chakraborty[2], and Samiran Chattopadhyay[2]

[1] Department of Computer Science and Engineering, Netaji Subhash Engineering College, Kolkata- 700 152, India
bose.shilpi08@gmail.com, chandradas@hotmail.com
[2] Department of Information Technology, Jadavpur University, Kolkata- 700 092, India
abir126@gmail.com, samiranc@it.jusl.ac.in

Abstract. Microarray experiments normally produce data sets with multiple missing expression values, due to various experimental problems. Unfortunately, many algorithms for gene expression analysis require a complete matrix of gene expression values as input. Therefore, effective missing value estimation methods are needed to minimize the effect of incomplete data during analysis of gene expression data using these algorithms. In this paper, missing values in different microarray data sets are estimated using different partition-based clustering algorithms to emphasize the fact that clustering based methods are also useful tool for prediction of missing values. However, clustering approaches have not been yet highlighted to predict missing values in gene expression data. The estimation accuracy of different clustering methods are compared with the widely used KNNimpute and SKNNimpute methods on various microarray data sets with different rate of missing entries. The experimental results show the effectiveness of clustering based methods compared to other existing methods in terms of Root Mean Square error.

Keywords: Microarray analysis, missing value estimation, c-means, fuzzy c-means, possibilistic c-means, fuzzy possibilistic c-means.

1 Introduction

Recent advancement of microarray technologies has made the experimental study of gene expression data faster and more efficient. Microarray techniques, such as DNA chip and high density oligonucleotide chip are powerful biotechnologies as they are able to record the expression levels of thousands of genes simultaneously [1].

The data generated in a set of microarray experiments are usually gathered in a matrix with genes in rows and experimental conditions in columns. Frequently, these matrices contain missing values (MVs). This is due to the occurrence of imperfections during the microarray experiment (e.g. insufficient resolution, spotting problems, deposition of dust or scratches on the slide, hybridization failures etc.) that create

N. Meghanathan et al. (Eds.): Advances in Computing & Inform. Technology, AISC 177, pp. 37–47.
springerlink.com

suspected values, which are usually thrown away and set as missing [2]. In large-scale studies involving thousands to tens of genes and dozens to hundreds of experiments, the problem of missing values may be severe. Virtually every experiment contains some missing entries and more than 90% of genes are effected. The presence of missing gene expression values constitutes a problem for downstream data analysis, since many of the methods employed, such as principal component analysis [3] or singular value decomposition [4] (e.g. classification and model-based clustering techniques) require complete matrices . Due to economic reasons or biological sample availability, repeating the microarray experiments in order to obtain a complete gene expression matrix is usually not feasible and also analysis results can be influenced by the estimation of replacing the missing values. Thus, in order to minimize the effect of missing values on analysis and avoid improper analysis, missing value estimation is an important preprocess.

Generally, the procedures for dealing with the randomly present missing data can be grouped into three categories [5], [2]: (1) Ignorance-based procedures: This is the most trivial approach to deal with data sets when the proportion of complete data is small, but the elimination brings a loss of information; (2) Model-based procedures: This is a missing data recovery method, which defines a model for the partially missing data. However, the complexity of the method prevents the applications of large data sets; (3) Imputation-based procedures: This is the type of missing data substitution methods, which fill the missing values by certain means of approximation. Statistical imputation belongs to this category, where the missing values are substituted by a statistically inspired value that has a high likelihood for the true occurrence, for example the mean values computed from the set of non-missing data records.

There are several simple ways to deal with missing values such as deleting genes with missing values from further analysis, filling the missing entires with zeroes, or imputing missing values of the average expression level for the gene (’row average’) [2] etc. Two advanced estimation methods for missing value estimation in microarray data have been proposed by Troyankaya et al. [5]; a weighted K-nearest neighbor method (KNNimpute) and a singular value decomposition method (SVDimpute). KNNimpute method is proposed as a robust and sensitive method for missing value estimation. It uses the KNN procedure to select genes, and uses weighted linear combinations to predict missing values. Recently, there is an estimation method called sequential K-nearest neighbor method (SKNNimpute) [6] for microarray data. This imputes missing values sequentially from the gene having least missing values and uses the imputed value for the latter imputation. Efficiencies of KNNimpute and SKNNimpute are better than the above mentioned simple methods in terms of missing value prediction error on non time series or noisy data. SVDimpute that takes all gene profile correlation information into consideration yields best results on time series data with low noise levels. However, estimation abilities of KNNimpute and SKNNimpute depend on the important model parameter K-value, the number of gene neighbor used to estimate the missing value. The parameter is usually specified by the user, which requires the user have some domain knowledge. There is no theoretical way, however, to determine these parameters appropriately. Several other methods have also been developed to estimate missing values. Bayesian principal component analysis (BPCA) [7] is shown to perform exceptionally well [8], [9]. However, BPCA

is a sophisticated method that is highly dependent on the number of principal axes [8]. The fixed-rank approximation algorithm (FRAA) proposed by Friedland et al. [10] carries out the estimation of all missing entries in the gene expression data matrix simultaneously based on the singular value decomposition (SVD) method. Local least-squares imputation (LLSimpute) by Kim et al. [11] exploits the local similarity structures in the data and uses the least-squares optimization method to find the missing values that are represented as a linear combination of similar genes. However, the prediction error generated using these methods still impacts on the performance of statistical and machine learning algorithms including class prediction, class discovery, and differential gene identification algorithms [12]. There is, thus, considerable potential to develop new techniques that will provide minimal prediction errors for different types of microarray data including both time and non-time series sequences.

Current research demonstrates that if the correlation/similarity between genes is exploited then missing value prediction error can be reduced significantly [13] in gene expression data. Cluster analysis [14], which partitions the given data set into distinct subgroups, is also applied to predict missing values in microarray data. Intuitively, objects in a cluster are more similar to each other than those belonging to different clusters. In this sense, objects in a cluster are more correlated with each other, whereas objects in different clusters are less correlated. As it can partitions different objects into groups, based on some similarity/dissimilarity criterion, it can also be used to discover structures based on similarity/dissimilarity in gene expression data without providing any interpretation. After clustering, missing values present in a gene can be predicted more accurately from other similar genes belonging to the same cluster.

In this paper, prediction accuracies are given for estimation of missing values in microarray gene expression data with respect to RMS error, using different partition based clustering algorithms. The effectiveness of the partition-based clustering methods, along with a comparison with SKNN and KNN imputation methods, is demonstrated on three microarray data sets.

2 Different Partition-Based Clustering Algorithms for Estimaion of Missing Values

In this section different partition-based clustering algorithms are described and then a new imputation method has been demonstrated to predict missing values in microarray gene expression data.

2.1 Notation

Throughout this paper, microarray data are represented by matrices with rows corresponding to genes and columns to experimental conditions. In particular, G represents original data matrix (with real MVs), while S is a complete gene expression matrix without any missing values with N genes and E experiments (with N >> E) after preprocessing G. In this S matrix, data are randomly deleted to create test data matrix T. X represents a set containing N number of genes. Every gene contains E number of attributes.

2.2 c-Means Clustering Algorithm

The algorithm proceeds by partitioning N number of objects into c nonempty subsets. During each partition, the centroids or means of the clusters are computed. This process iterates until the criterion function converges. Typically, the square-error criterion is used, defined as

$$E = \sum_{i=1}^{K} \sum_{x_k \in U_i} |x_k - m_i|^2 \tag{1}$$

The main steps of the c-means algorithm [15] are as follows:

1) Arbitrarily choose c number of object from X and they are assigned in m_i, i = 1 to c as initial cluster means.
2) Assign each data object x_k to the cluster U_i for the closest mean.
3) Compute new mean for each cluster using

$$m_i = \frac{\sum_{x_k \in U_i} x_k}{|U_i|} \tag{2}$$

where $|U_i|$ is the number of objects in cluster U_i.
4) Iterate until criterion function converges, i.e., there are no more new assignments.

2.3 Fuzzy c-Means (FCM) Clustering Algorithm

This is a fuzzification of the c-means clustering algorithm. It partitions a set of N objects $\{x_k\}$ into c clusters by minimizing the objective function

$$J = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^p ||x_k - m_i||^2 \tag{3}$$

where $1 \leq p < 1$ is the fuzzifier, m_i is the i^{th} cluster center, $\mu_{ik} \in [0, 1]$ is the membership of the k^{th} pattern and ||.|| is the distance norm, such that

$$m_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^p x_k}{\sum_{k=1}^{N} (\mu_{ik})^p} \tag{4}$$

and

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{p-1}}} \tag{5}$$

$\forall i, d_{ik} = ||x_k - m_i||^2$, subject to $\sum_{i=1}^{c} \mu_{ik} = 1, \forall k,$ and $0 < \sum_{k=1}^{N} \mu_{ik} < N, \forall i.$

The algorithm [16] proceeds as follows:

1) Pick the initial means m_i, i = 1, · · · , c. choose value for fuzzifier p and threshold ε. Set the iteration counter t = 1.
2) Repeat Steps 3-4, by incrementing t, until $|\mu_{ik}(t) - \mu_{ik}(t-1)| > \varepsilon$.
3) Compute μ_{ik} by eqn. (5) for c clusters and N data objects.
4) Update means m_i by eqn. (4).

2.4 Possibilistic c-Means (PCM) Clustering Algorithm

It partitions a set of N objects $\{x_k\}$ into c clusters by minimizing the objective function

$$J = \sum_{i=1}^{c}\sum_{k=1}^{N}(t_{ik})^q ||x_k - m_i||^2 + \sum_{i=1}^{c}\eta_i \sum_{k=1}^{N}(1 - t_{ik})^q \tag{6}$$

where $1 \leq q < 1$ is the fuzzifier, m_i is the ith cluster center, $t_{ik} \in [0, 1]$ is the typical membership of the k^{th} pattern, η_i are suitable positive integers and ||.|| is the distance norm, such that

$$m_i = \frac{\sum_{k=1}^{N}(\mu_{ik})^q x_k}{\sum_{k=1}^{N}(t_{ik})^q} \tag{7}$$

and

$$t_{ik} = \frac{1}{1 + \left(\frac{d_{ik}^2}{\eta_i}\right)^{\frac{1}{q-1}}} \tag{8}$$

and

$$\eta_i = K\frac{\sum_{k=1}^{N} t_{ik}^q d_{ik}^2}{\sum_{k=1}^{N} t_{ik}^q} \tag{9}$$

here typically K is chosen to be 1. The main steps of the PCM algorithm [17] are as follows:

1) Pick the initial means m_i, i = 1, · · · , c. choose value for fuzzifier p and threshold ε. Set the iteration counter it = 1.
2) Repeat Steps 3-4, by incrementing it, until $|t_{ik}(it) - t_{ik}(it-1)| > \varepsilon$.
3) Compute t_{ik} by eqn. (8) for c clusters and N data objects.
4) Update means m_i by eqn. (7).

2.5 Fuzzy-Possibilistic c-Means (FPCM) Clustering Algorithm

It partitions a set of N objects $\{x_k\}$ into c clusters by minimizing the objective function

$$J = \sum_{i=1}^{c} \sum_{k=1}^{N} \left(\mu_{ik}^{p} + t_{ik}^{q}\right) ||x_k - m_i||^2 \tag{10}$$

subject to the constraints p > 1, q > 1, $0 \leq \mu_{ik}, t_{ik} \leq 1$, and

$$\sum_{i=1}^{c} \mu_{ik} = 1, \forall k \tag{12}$$

and

$$\sum_{k=1}^{N} t_{ik} = 1, \forall i \tag{12}$$

and

$$m_i = \frac{\sum_{k=1}^{N} \left(\mu_{ik}^{p} + t_{ik}^{q}\right) x_k}{\sum_{k=1}^{N} \left(\mu_{ik}^{p} + t_{ik}^{q}\right)} \tag{13}$$

here μ_{ik} is the fuzzy membership value given in eqn. (5) and t_{ik} is the typical or possibilistic membership value given in eqn.(8), p and q are fuzzifiers.

The main steps of the FPCM algorithm [18] are as follows:

1) Pick the initial means m_i, i = 1, · · · , c. choose value for fuzzifier p, q and threshold ε. Set the iteration counter it = 1.
2) Repeat Steps 3-4, by incrementing it, until $|\mu_{ik}(it) + t_{ik}(it) - \mu_{ik}(it - 1) + t_{ik}(it - 1)| > \varepsilon$.
3) Compute μ_{ik} by eqn. (5) and t_{ik} by eqn. (8) for c clusters and N data objects.
4) Update means m_i by eqn. (13).

2.6 Imputation of Missing Values

Initially, all missing values in T are replaced by the estimation given by row (gene) averages to obtain a complete matrix. Specially, this step of gene average substitution, performed in all clustering methods, provides the possibility of contributing the maximum number of genes for estimating the missing values. Then any one of the above mentioned clustering algorithms are executed on this complete matrix. The missing values are imputed by making use of the weighted mean of the values of the corresponding attribute over all clusters. The weighting factors are the membership degrees u_{ik} of a gene x_k to the i^{th} cluster. The missing gene expression value x_{kj} is imputed by:

$$x_{kj} = \frac{\left(\sum_{i=1}^{c} u_{ik}^{l} v_{ij}\right)}{\sum_{i=1}^{c} u_{ik}^{l}} \tag{14}$$

where u_{ik} is the membership value of k^{th} gene in the i^{th} cluster. v_{ij} represents value of j^{th} attribute of mean of i^{th} cluster and l is the fuzzifier. For hard c-mean clustering membership values are either 0 or 1.

The main steps of the imputation algorithm is as follows:

1) Initially all missing values in T are replaced by the estimation given by row (gene) averages for obtaining a complete matrix.
2) Apply any one of the above mentioned clustering algorithm to cluster genes.
3) Estimate missing values by using eqn.(14) with the means obtained from clustering result.
4) Repeat steps 1 and 2 for different number of clusters.

3 Experimental Results

The above mentioned different partition-based clustering algorithms are compared with the previously developed KNNimpute and SKNNimpute methods by imputation of microarray data. Data sets used in this work are selected from publically available microarray data. Three microarray data sets are used: cluster analysis and display of genome-wide expression patterns (data 1) [19], Genomic expression programs in the response of yeast cells to environmental changes (data 2) [20] and the transcriptional program in the response of human fibroblast to serum (data 3)[21]. The metric used to assess the accuracy of estimation is Root Mean Squared (RMS) error which is calculated as follows:

$$RMS_{error} = \frac{\sum_{h=1}^{n} (R_h - I_h)^2}{n} \tag{15}$$

where R_h is the real value, I_h is the imputed value, and n is the number of missing values.

Before any further process, each data set is preprocessed for the evaluation, by removing rows and columns containing missing expression values greater than 50% and rest are replaced by row average values, yielding complete matrices. For every data set between 1 and 20% of the data are deleted at random to create test data set. Each method is then used to recover the introduced missing values for each data set, and the estimated values are compared to those in the original data set.

Every clustering method is executed for c = 5 to 50, where c is the number of clusters. The experiments show that for c > 50 the clustering results detoriates. The value of fuzzifier is varied from 1.2 to 2. For every clustering method best result (i.e. minimum RMS error) is taken for different values of fuzzifier as well as for different values of number of clusters (c).The result is shown for different rate of missing entries present in every data set.

The efficiency of different partition-based clustering methods mentioned here are compared with the KNNimpute and the SKNNimpute methods by applying them to three microarray data sets with different missing rates. Both KNNimpute and SKNNimpute methods require the value of k which is the number of nearest neighbors used in imputation. When k is between 5 and 20, they have given good performances. Accordingly, minimal RMS errors of these two methods are shown by

varying k between 5 to 20 in every data set with different rates of missing values. In Table 1, prediction accuracies of different clustering methods are shown for different number of clusters. In Table 2, only best results are shown for data 2 and Data 3 using different clustering algorithms.

In figure 1, it is found that c-mean has given best results compared to all other partition-based clustering algorithms mentioned here and also with respect to KNNimpute and SKNNimpute methods for all different rates of missing entries in data 1 and data 2. FCM, PCM, and FPCM clustering methods also have given better results with respect to KNNimpute and SKNNimpute methods for all cases in data 1 and data 2. For data 3, FCM gives best results for all rates of missing. The other clustering methods have also given better results compared to KNNimpute and SKNNimpute methods for data 3.

Table 1. Comparative Performance Analysis of Different Clustering Methods on Data 1

Rate of data missing (%)	No. of clusters	Prediction Accuracy			
		c-Mean	FCM	PCM	FPCM
1	5	0.484	0.523	0.519	0.523
	10	0.474	0.522	0.524	0.483
	20	0.459	0.502	0.518	0.448
	30	0.45	0.499	0.522	0.454
	50	0.437	0.563	0.523	0.443
5	5	0.534	0.531	0.533	0.543
	10	0.47	0.529	0.529	0.494
	20	0.46	0.527	0.527	0.463
	30	0.452	0.502	0.525	0.464
	50	0.441	0.503	0.525	0.503
10	5	0.544	0.555	0.543	0.531
	10	0.501	0.524	0.513	0.494
	20	0.474	0.523	0.513	0.473
	30	0.463	0.515	0.524	0.502
	50	0.441	0.564	0.535	0.532
20	5	0.583	0.576	0.587	0.594
	10	0.556	0.563	0.542	0.542
	20	0.494	0.547	0.542	0.502
	30	0.463	0.542	0.564	0.524
	50	0.468	0.548	0.564	0.524

Table 2. Best Performance of Different Clustering Methods on Data 2 and Data 3

Data Set	Rate of data missing(%)	Prediction Accuracy							
		No. of clusters	c-Mean	No. of clusters	FCM	No. of clusters	PCM	No. of clusters	FPCM
Data 2	1	50	0.51	30	0.59	10	0.7	20	0.61
	5	50	0.53	30	0.62	10	0.71	10	0.61
	10	50	0.54	25	0.64	20	0.73	10	0.62
	20	50	0.55	30	0.65	20	0.74	20	0.65
Data 3	1	30	0.74	15	0.68	50	0.8	10	0.73
	5	20	0.75	20	0.7	15	0.8	25	0.76
	10	30	0.76	10	0.71	10	0.85	30	0.75
	20	30	0.8	10	0.75	10	0.86	30	0.8

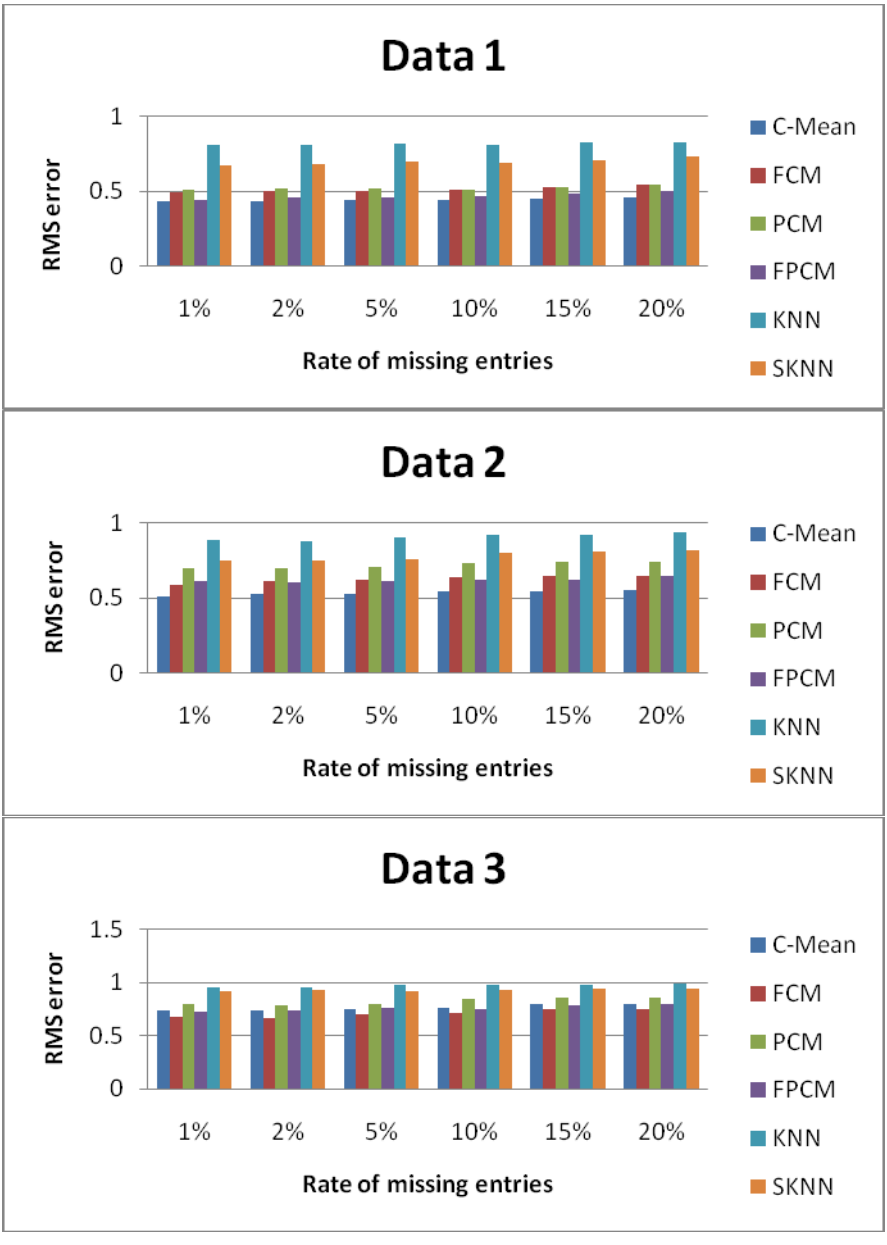

Fig. 1. Comparison of accuracy of Different clustering methods with KNNimpute and SKNNimpute methods for three types of data sets over 1 to 20% data missing. The accuracies are evaluated by RMS error.

4 Conclusion

In this paper, the performance accuracy of different partition-based clustering algorithms for missing value estimation in microarray data sets are compared with KNNimpute and SKNNimpute methods. The experimental results show that in all cases clustering methods have given better results than KNNimpute and SKNNimpute methods in terms of RMS error. So, it can be concluded that clustering methods are also very effective for missing value estimation in microarray gene expression data.

References

1. Schulze, A., Downward, J.: Navigating gene expression using microarrays - a technology review. Nat. Cell Biol. 3, E190–E195 (2001)
2. Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., Powell, J.I., Yang, L., Marti, G.E., Moore, T., Hudson, J.J., Lu, L., Lewis, D.B., Tibshirani, R., Sherlock, G., Chan, W.C., Greiner, T.C., Weisenburger, D.D., Armitage, J.O., Warnke, R., Staudt, L.M.: Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403, 503–511 (2000)
3. Raychaudhuri, S., Stuart, J.M., Altman, R.B.: Principal component analysis to summarize microarray experiments: application to sporulation time series. In: Pac. Symp. Biocomputing, pp. 455–466 (2000)
4. Alter, O., Brown, P.O., Bostein, D.: Singular value decomposition for genome-wide expression data processing and modeling. Proc. Natl Acad. Sci. USA 97, 10101–10106 (2000)
5. Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Bostein, D., Altman, R.B.: Missing value estimation methods for DNA microarrays. Bioinformatics 17, 520–525 (2001)
6. Kim, K.Y., Kim, B.J., Yi, G.S.: Reuse of imputed data in microarray analysis increases imputation efficiency. BMC Bioinformatics 5(160) (2004)
7. Oba, S., Sato, M.A., Takemasa, I., Monden, M., Matsubara, K.I., Ishii, S.: A bayseian missing value estimation method for gene exression profile data. Bioinformatics 19, 2088–2096 (2003)
8. Wang, X., Li, A., Jiang, Z., Feng, H.: Missing value estimation for DNA microarray gene expression data by support vector regression imputation and orthogonal coding scheme. BMC Bioinformatics 7, 1–10 (2006)
9. Wong, D.S.V., Wong, F.K., Wood, G.R.: A multi-stage approach to clustering and imputation of gene expression profiles. Bioinformatics 23, 998–1005 (2007)
10. Friedland, S., Niknejad, A., Chihara, L.: A simultaneous reconstruction of missing data in DNA microarrays. Linear Algebra Appl. 416, 8–28 (2006)
11. Kim, H., Golub, G.H., Park, H.: Missing value estimation for DNA microarray gene expression data: local least squares imputation. Bioinformatics 21, 187–198 (2005)
12. Sehgal, M.S.B., et al.: Statistical neural networks and support vector machine for the classification of genetic mutations in ovarian cancer. In: IEEE CIBCB 2004, USA (2004)
13. Sehgal, M.S., et al.: K-ranked covarience based missing values estimation for microarray data classification. In: HIS (2004)

14. Au, W.-H., Chan, K.C.C., Wong, A.K.C., Wang, Y.: Attribute clustering for grouping, selection, and classification of gene expression data. IEEE Trans. on Computational Biology and Bioinformatics 2(2) (2005)
15. Tou, J.T., Gonzalez, R.C.: Pattern recognition principles. Addison-Wesley, London (1974)
16. Bezdek, J.C.: Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York (1981)
17. Krishnapuram, R., Keller, J.: A possibilistic approach to clustering. IEEE Trans. Fuzzy Syst. 4(3), 393–396 (1993)
18. Pal, N.R., Pal, K., Bezdek, J.C.: A mixed c-means clustering model. In: IEEE Int. Conf. Fuzzy Systems, Spain, pp. 11–21 (1997)
19. Eisen, M., Spellman, P., Brown, P., Bostein, D.: Cluster analysis and display of genome wide expression patterns. Proc. Natl Acad. Sci., USA 95, 14863–14868 (1998)
20. Gasch, A., Spellman, P., Kao, C., Carmel-Harel, O., Eisen, M., Storz, G., Bostein, D., Brown, P.: Genomic expression programs in the response of yeast cells to environmental changes. Mol. Biol. Cell. 11, 4241–4257 (2000)
21. Iyer, V.R., Eisen, M.B., Ross, D.T., Schuler, G., Moore, T., Lee, J.C.F., Trent, J.M., Staudt, L.M., Hudson, J.J., Bogosk, M.S., et al.: The transcriptional program in the response of human fibroblast to serum. Science 283, 83–87 (1999)