

Phimmarin Keerin and Werusak Kurutach
Faculty of Information Science and Technology
Mahanakorn University of Technology
Bangkok, Thailand
email: p.keerin@gmail.com, werusak@mut.ac.th

Abstract—Gene expressions measured using microarrays usually encounter the problem of missing values. Leaving this unsolved may critically degrade the reliability of any consequent downstream analysis or medical application. Yet, a further study of microarray data might be impossible with many analysis methods requiring a complete data set. This paper introduces a new methodology to impute missing values in microarray data. The proposed algorithm, CKNN impute, is an extension of k nearest neighbor imputation with local data clustering being incorporated for improved quality and efficiency. Gene expression data is typically represented as a matrix whose rows and columns correspond to genes and experiments, respectively. CKNN kicks off by finding a complete dataset via the removal of rows with missing value(s). Then, k clusters and their corresponding centroids are obtained by applying a clustering technique on the complete dataset. A set of similar genes of the target gene (with missing values) are those belonging to the cluster, whose centroid is the closest target. Having known this, the target gene is imputed by applying k nearest neighbor method with similar genes previously determined. Empirical evaluation with published gene expression datasets suggest that the proposed technique performs better than the classical k nearest neighbor method and its extension found in the literature.

ments, which can be expensive and time consuming. Note also that the repeat of experiments may not guarantee data completeness.

The dilemma of missing values is an actual yet challenging issue confronted in many researches in machine learning and data mining. Missing values may generate bias and negatively affect the quality of the supervised learning process or the performance of data clustering algorithm [22], [23]. In general, dealing with missing values is to find an approach with the ability to fill in these previously unknown whilst maintaining (or approximate as closely as possible) the original distribution of the data under examination. Imputation is a term that denotes a procedure of replacing the missing values in a dataset by some plausible values. Many analysis methods, such as principle component analysis [9] or clustering analysis, require a complete data matrix. Missing data are often replaced by zeros or less often, by an average expression over the row, or ‘row average’ [1]. These statistical approaches are rarely optimal, since they do not take into consideration the correlation structure represented within the data.

In response, a number of more intelligent techniques, such as k-nearest neighbors (KNN) [18], local least squares (LLS) [13] and Bayesian principal-component analysis (BPCA) [11], have been recently introduced in the literature, especially for microarray data (see [1] for a survey). In many studies that investigate the effect of missing values on biomarker discovery and disease classification, the empirical findings suggest minor differences among KNN, LLS and BPCA with respect to the classification accuracy [21]. As compared to KNN, BPCA and LLS are less efficient and not applicable to a large dataset [19]. In general, the KNN impute method find k other genes with expressions most similar to that of target gene (with missing values). The missing value of target gene is estimated by the average of corresponding values recorded for those k closest genes. To boost the performance, several extensions of KNN, so-called weighted KNN [10], [12], are put forward with different weighting schemes for nearest neighbors. Specifically, the contribution of each gene is weighted by the similarity between its expression and that of target gene. As such, the higher the similarity is, the greater the height becomes.

Despite its simplicity and the reported success, the acquisition of nearest neighbors, for both KNN and the weighted

DNA microarray technology is a method for probing the expression of thousands of genes under a variety of experimental conditions. Gene expression data obtained from microarray experiments has inspired several applications, including the identification of differentially expressed genes for molecular studies or drug therapy response [20] and the creation of classification systems for improved cancer diagnosis [15]. Traditional data mining algorithms such as k -means and agglomerative hierarchical clustering have proven useful for identifying biologically relevant clusters of tissue samples/experimental conditions and genes [7].

Despite such benefits, gene expression data are frequently peppered with missing values (MVs) that may occur for a variety of reasons. These include insufficient resolution, image corruption, array fabrication error, and experimental error during the laboratory processing [11]. Therefore, the treatment of missing values is a critical step in preprocessing as data quality is a major concern in consequent downstream analysis. This so-called data cleaning [6] the need for repeating experi-

counterpart, is driven only by proximity. This is effective when neighbors are spatially distributed, i.e., groups/clusters of neighbors do not exist. Such an assumption and the consequent selection of neighbors can be sub-optimal with most datasets in which data clusters normally reside. For the imputation of gene expression values, this may lead to an inaccurate estimation, especially for those target genes situate by the edge of a cluster. Hence, bringing information regarding data clusters into the imputation process may resolve the aforementioned problem. This paper presents a novel extension of KNN, Cluster based KNN (CKNN) impute, which is based on the discovered clustering model. The k -means clustering algorithm is used to obtain clusters of dataset under examination. Instead of using all available genes, only those genes in the cluster whose centroid is the closest to the target gene are candidates for the nearest neighbors. To prove the underlying intuition, this new method is evaluated on several published microarray datasets, in comparison with basic statistical techniques, the traditional KNN and its weighted variation. The remaining of this paper is organized as follow. In Section II, details of related work are included to provide background and basic concepts relating to missing value imputation. Section III presents the CKNN imputation algorithm, while the performance evaluation is explained in Section IV. The paper is concluded in Section V with the perspective of future work.

II. RELATED WORK

A. DNA Microarray

Bioinformatics has been one of the most fast developing fields in scientific research. As an important tool in bioinformatics, microarray technology was developed in a great need. It provides a remarkable methodology to monitor the expression level of a large amount of genes. This technology has created a number of new opportunities to study human disease. For instance, many microarray studies have discovered clinical subtypes of cancer with different prognoses from those identified by other methods [2]. Many biologists have performed microarray experiments on various problem domains. However, the resulting mass of data, often as millions of measurements, can be overwhelming for human comprehension and pose great challenges on data preprocessing and analysis methodology.

One of the major difficulties with microarray data is the presence of missing values that are commonly due to a number of diverse reasons. These include insufficient resolution, image corruption, dust or scratches on the slide or experimental error during the laboratory process. Moreover, missing data may occur systematically as a result of the robotic methods used to create them [14]. Generally, missing of data around 1-10% would affect up to 95% of the genes in any microarray experiment [4]. Of course, one possible solution to recover the dataset with missing values is to repeat the experiment, which is inefficient and sometimes infeasible. Hence, the missing values in the original data need to be predicted mathematically. For a reliable downstream analysis and application, it

is desirable to develop a proper algorithm to achieve the maximum accuracy of imputation for microarray data.

B. K-Nearest Neighbor (KNN) Imputation

KNN is a standard missing value imputation method, which has been introduced by [18]. This nearest-neighbor based technique takes advantage of the correlation structure in microarray data by selecting a set of genes with expression profiles similar to the gene of interest to impute missing values. Accordingly, the imputation process is typically divided into two steps. In the first step, a set of genes nearest to the gene with missing values (i.e., target gene) are selected. Let microarray data be formulated as a gene expression matrix G with m rows and n columns corresponding to genes and experiments, respectively. To impute a missing value x_{gi} of gene $x_g, g \in \{1, \dots, m\}$ and experiment $i \in \{1, \dots, n\}$, it is to find k other genes, each with a known value for experiment i and its expression profile (considering all experiments) being the most similar to that of gene x_g .

A number of different measures, namely Pearson correlation, Euclidean distance and variance minimization, have been investigated in [18] for the determination of gene similarity. In spite of its sensitivity to outliers, Euclidean distance appears to be an accurate norm. In principle, Euclidean distance is defined as follows.

$$d_{ij} = \text{dist}(x_i, x_j) = \sqrt{\sum_{p=1}^n (x_{ip} - x_{jp})^2}, \quad (1)$$

where $\text{dist}(x_i, x_j)$ denotes the Euclidean distance between genes x_i and x_j , n is the number of microarray experiments, and x_{ip} is the p^{th} experiment of gene x_i .

The second step involves the prediction of the missing value using the observed values belonging to the selected group of genes. At this stage, an average of values in experiment i from the k closest genes is then used as an estimate for the missing value in gene x_g . The estimated value $\tilde{x}_{gi}$ of the missing x_{gi} can be calculated by:

$$\tilde{x}_{gi} = \frac{\sum_{\alpha \in N_g} x_{\alpha i}}{k}, \quad (2)$$

here N_g is the set of k nearest neighbors of gene x_g . In the weighted variation, the contribution of each gene $x_{\alpha} \in N_g$ is weighted by the similarity of its expression to that of gene x_g . Intuitively, higher weights are assigned to more similar gene.

A weighted average of values from k nearest genes is then used as an estimate for the missing value in target gene. This weight calculation is defined by:

$$\tilde{x}_{gi} = \sum_{\alpha \in N_g} x_{\alpha i} w_i; \quad w_i = \frac{1/d_i}{\sum_{i=1}^k 1/d_i}, \quad (3)$$

where k is the number of closest genes and d_i is the distance between i^{th} gene and target gene.

C. K-Means Clustering

K-means is one of the simple and highly efficient unsupervised learning algorithms that solve the well-known clustering problem. It aims to divide a given set of data into a predefined number of groups or clusters. Having accomplished this, a set of centroids (i.e., cluster centers) is systematically discovered and later used as future reference. For a given gene expression data $G \in \mathbb{R}^{m \times n}$, each x_{ig} , $g = 1 \dots m$, corresponds to a gene which can be represented by a profile of n -experiment values, i.e., $x_g = (x_{g1}, \dots, x_{gn})$. The k -means algorithm searches for the partition $\pi = \{C_1, \dots, C_k\}$ of genes $(x_1, \dots, x_m)$ into k clusters, such that genes in the same cluster are more similar to each other than to those in the other clusters. This is achieved through minimizing the following objective function:

$$J(U, Z) = \sum_{i=1}^k \sum_{j=1}^m \sum_{l=1}^n u_{il} (x_{ij} - z_{lj})^2, \quad (4)$$

where $\sum_{i=1}^k u_{il} = 1$, $U \in \mathbb{R}^{m \times n}$ is a matrix in which each entry u_{il} represents a membership degree that object $x_i \in G$ has with cluster $C_l \in \pi$ ($u_{il} \in \{0, 1\}$ and $u_{il} \in [0, 1]$ for crisp and soft clustering, respectively). In addition, $Z = \{z_1, \dots, z_k\}$ denotes a set of vectors each representing the centroid of k clusters, i.e., $z_l = (z_{l1}, \dots, z_{ln})$, $\forall l = 1 \dots k$. This algorithm is composed of the following steps:

- Step1: randomly select an initial set of k centroids, i.e. Z .
- Step2: assign each object to the cluster whose centroid is the closest, i.e. update U .
- Step3: when all objects have been assigned, recalculate the positions of the k centroids, i.e. update Z .
- Step4: repeat steps 2 and 3 until the centroids are no longer changed.

Although it can be proved that the aforementioned procedure will always terminate, k -means does not necessarily find the most optimal configuration, corresponding to the objective function $J(U, Z)$. In addition, the algorithm is also sensitive to the initial randomly-selected centroids [7].

D. Davies-Bouldin(DB) Measure

The quality of a data partition, i.e. a set of clusters, can be judged using the validity criterion such as compactness of data within a cluster and separation between clusters [8]. One of the effective indices is DB measure of [3] that has proven to be a reliable tool for differentiating the clusterings generated by different methods, or multiple runs of an instable algorithm like k -means. This measure of a partition π can be defined by the following.

$$DB(\pi) = \frac{1}{k} \sum_{\forall C_a, C_b \in \pi} \frac{\delta_a + \delta_b}{d(z_a, z_b)}, \quad (5)$$

where k denotes the number of clusters in a partition π , to which C_a and C_b belong. Note that $d(z_a, z_b)$ is the distance between centroids z_a and z_b of clusters C_a and C_b , respectively. Also, δ_a is the average of distances between all objects in cluster C_a and the centroid z_a . Small values of DB correspond to clusters that are compact, and whose centroids are far-away

from each other. Consequently, the number of clusters that minimizes this DB measure can be taken as the optimal number of clusters.

III. PROPOSED METHOD

In this section, the k nearest neighbor imputation that depends on a clustering model will be introduced. This CKNN method includes three major steps of finding a number of clusters, generating a clustering model that will be used as a reference, and apply the KNN impute in conjunction with the discovered clusters. Each of these is described in the following sections.

A. Getting a Number of Clusters

Clustering algorithms is unsupervised learning processes, i.e., users are usually required to set some working parameters. These algorithmic variables vary from one technique to another, but most clustering method requires a parameter that either directly or indirectly specifies the number of cluster. One of the typical decisions regards the number of clusters (k). This is also the case for k -means, which will be employed in this research for its simplicity and efficiency. Setting the value of k requires either detailed prior knowledge of the investigated data, or time-consuming trial and error. For the latter, a user must have sufficient domain knowledge to know what a good clustering looks like. However, if the data set is very large or of high dimensionality, human verification could become difficult. It is necessary to have an algorithm that can efficiently determine a reasonable number of clusters to return from any clustering algorithm. The initial step of proposed algorithm is identified the correct number of clusters returned by k -means. This procedure can be summarized below.

- First, get the data matrix $G' \in \mathbb{R}^{m' \times n}$ from the original gene expression data $G \in \mathbb{R}^{m \times n}$, by removing rows with missing values. Having done this, G' is hereafter referred to as a complete data matrix, with $m' \leq m$.
- Clustering a complete data matrix G' using the k -means algorithm. The objective function of this clustering process is

$$J(U, Z) = \sum_{i=1}^k \sum_{j=1}^{m'} \sum_{l=1}^n u_{il} (x_{ij} - z_{lj})^2, \quad (6)$$

Generate a set of data partitions $\{\pi_2, \pi_3, \dots, \pi_\beta\}$ using different values of $k \in \{2, 3, \dots, \beta\}$. In this research, $\beta=15$ is used for efficiency. It is noteworthy that a more general heuristics like $\beta = \sqrt{m}$ of [7] can be applied, however with more computational time.

- Then, find the data partition from $\{\pi_2, \pi_3, \dots, \pi_\beta\}$ with the lowest DB measure. Consequently, the number of clusters k that minimizes DB is taken as the optimal number of clusters for the next processing stage.

B. Obtaining a Clustering Model

Having known a good number of clusters (k), k -means is again used with G' to derive a clustering model π for the imputation. Note that the partition π consists of k clusters

$\{C_1, \dots, C_k\}$, each of which is represented by a unique centroid z_t , $t = 1 \dots k$. After that, i is to find the cluster with which any row x_g^* with missing values should be associated. Note that x_g^* is removed as the complete matrix G' is created, i.e. $x_g^* \in G, x_g^* \notin G'$. Such a row in gene expression data is called a target gene whose missing values will be estimated.

A target gene x_g^* is assigned to a cluster C_a only when

$$z_a = \operatorname{argmin}_{p=1 \dots k} d(z_p, x_g^*), \quad (7)$$

where z_p denotes the centroid of cluster C_p . To calculate the distance $d(z_p, x_g^*)$, it is necessary to fill in missing values of x_g^* such that both vectors are equally of n dimensions. For this purpose, the row average is applied to make a reasonable guess. The cluster membership discovered here will later be utilized for the imputation of the underlying missing values.

C. Applying Cluster-Based KNN Imputation

The k nearest neighbor (KNN) imputation method has proven simple and effective in the literature, whilst being generally competitive to other advanced techniques. It also requires less computational resources as compared to those more complex counterparts. However, the efficiency of KNN is still subjected to the number of genes (m), with the time complexity of searching for nearest neighbors being around $O(m^2)$. In other words, KNN does not scale up well with a very large set of data.

In order to increase to efficiency of imputation missing values in microarray data, the idea is to reduce biases and increase correlation of data by clustering method before imputing the missing values. With a clustering model, the neighbor search is restricted only to the cluster that the target gene is closest related. Thus, the time complexity would reduce dramatically to $O(q^2)$, where q is the average size (or number of genes) of clusters and $q \ll m$. The cluster-based KNN (CKNN) imputation algorithm is summarized as follows:

- Step1: find the complete data matrix $G' \in \mathbb{R}^{m' \times n}$ from the original gene expression data $G \in \mathbb{R}^{m \times n}$ by removing row with missing values.
- Step2: clustering the data matrix G' using k -means and different numbers of k .
- Step3: select the optimal value k by finding one of $\beta - 1$ partitions from step2, with the lowest DB measure.
- Step4: create a clustering model $\pi = \{C_1, C_2, \dots, C_k\}$ for further analysis, using k -means and the value of k found in the previous step.
- Step5: for target each gene (or row) with missing values, $x_g^* \in G, x_g^* \notin G'$:
 - Find the closest cluster C_* in π to the target gene.
 - Search for the nearest neighbors x_g of the target gene, provided that $x_g \in C_*$.
 - Apply KNN imputation to estimate the missing value of the target gene.

D. Parameter set

In Cluster-Based KNN imputation, the number of nearest neighbor, k , must be selected. Troyanskaya *et al.* [18] addressed this question in KNN imputation, with the best result being observed when k in the range 10-20. This finding is applicable when finding nearest neighbors from the entire population. However, smaller k is expected to be effective for CKNN where the search for nearest neighbors is pursued only in a definite cluster or subset of population. Hence, in this paper, $k = 3, 5, 8, 10, 15$ and 20 are explored.

IV. PERFORMANCE EVALUATION

This section presents the empirical study of the proposed method on several publicly available datasets, in comparison with related algorithms found in the literature.

A. Experimental Design

In this evaluation, six datasets from published microarray experiments are investigated. Four of these is obtained from the study of cell cycle regulate in *Saccharomyces cerevisiae* [16]. The first is generated from a complete data matrix of 4,304 genes and 18 experiments (Alpha). Missing values have been inserted with the purpose of being used to assess missing value estimation methods. The second data set is created from elutriation data presented as a complete matrix of 4,304 genes and 14 experiments (Elu). The third data set is from 784 cell cycle regulate genes and consists of 2,856 genes and 14 experiments (cyc.a). The fourth is of 242 genes and 14 experiments (cyc.b).

The fifth data set is acquired from the study of response to environmental changes in yeast [5]. It contains 5,431 genes and 13 experiments (Evn). And, the sixth is the cDNA microarray data set relevant to human colorectal cancer (ta.crc) [17]. This data set contains 758 genes and 205 primary CRCs including 127 non-metastatic primary CRCs, 54 metastatic primary CRCs to the liver, 24 metastatic primary CRCs to distance organs exclusive of the liver, and 12 normal colonic epithelia. Note that, for this evaluation, missing values are random created for each dataset in accordance with the prespecified levels of 1%, 2%, 3%, 4%, 5%, 10%, 15% and 20%. For the data matrix with dimension of $6,178 \times 18$ and 1% missing rate, there will be 1,112 missing values arbitrarily distributed over the underlying matrix.

The performance of the missing value imputation is evaluation by Normalized Root Mean Square Error (NRMSE), which is defined by:

$$NRMSE = \frac{\sqrt{\operatorname{mean}(x_{quess} - x_{ans})^2}}{\operatorname{std}(x_{ans})}, \quad (8)$$

where x_{quess} and x_{ans} are imputed values and known answer values presented in the original data matrix, respectively. In addition, $\operatorname{std}(x_{ans})$ is the standard deviation of known answer values. The lower NRMSE is, the better the value estimated by a computerized method.

B. Experimental Results

The NRMSE measures of investigated methods on alpha dataset are shown in Fig 1. These techniques include two statistical techniques of row average (row) and zero-value (zero). The conventional k nearest neighbors (knn) and the weighted extension (wknn) are presented as the base lines. In addition, cknn and wcknn denote two cluster-based KNN, with the first using a simple KNN for the imputation step and the other using the weighted KNN. To generalize this empirical study, k -mean that denotes the use of the closest centroid found with cknn as the estimate for missing value, is also investigated.

As shown in this figure, the cluster-based KNN imputation methods perform consistently better than the two basic statistical models. They also improve the quality of their base line, i.e. KNN and weighted KNN. Also, the weighted version of CKNN is slightly more efficient than the unweighted model. This is in line with the behavior of those basic KNN methods. Again, the proposed models are more effective than the simple use of centroid, presented as k -mean in this figures. The quality of guess answers degrades as the missing rate increases. Despite this, the aforementioned observations still hold true. These results are obtained using the size of nearest neighbors of 10 and 3, for the classical KNN and CKNN, respectively. Similar results can also be observed on other datasets, see details in Figs 2-6.

V. CONCLUSION

This paper proposes a novel version of k nearest neighbor imputation method to estimate the missing values in microarray data. The practice of k -mean clustering is uniquely applied to the problem of missing value imputation. The experimental results demonstrate the strength of this method. The performance of the proposed algorithm and related models are gauged based on the widely-used NRMSE analysis. It is discovered that the KNN imputation algorithm based on k -means method outperforms the normal KNN impute method, which is a simple and common approach for missing value imputation.

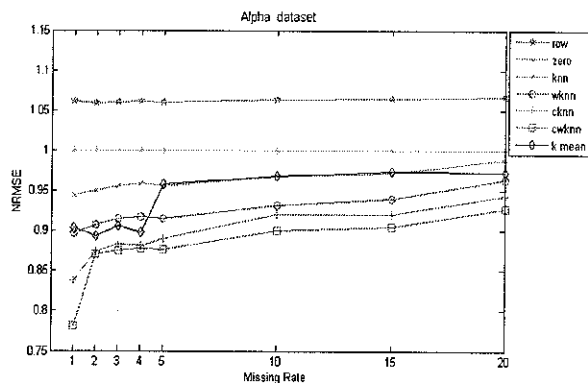


Fig.1. Experimental results on alpha dataset, NRMSE measures of investigated methods at different missing rates.

The empirical study also shows that the overall performance of the KNN based on clustering method is better than the basic KNN method, especially when the percentage of missing values is high. An interesting future work is found an effective weighting scheme to further improve the model introduced herein. In addition, the use of clustering model to help identifying nearest neighbors might be useful for other imputation algorithms that rely on such information for the calculation of estimated values.

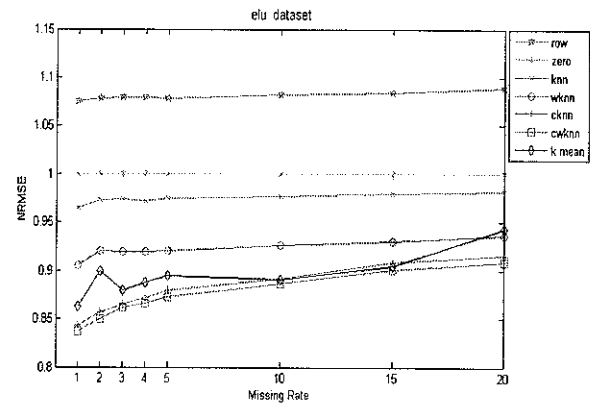


Fig.2. Experimental results on elu dataset, NRMSE measures of investigated methods at different missing rates.

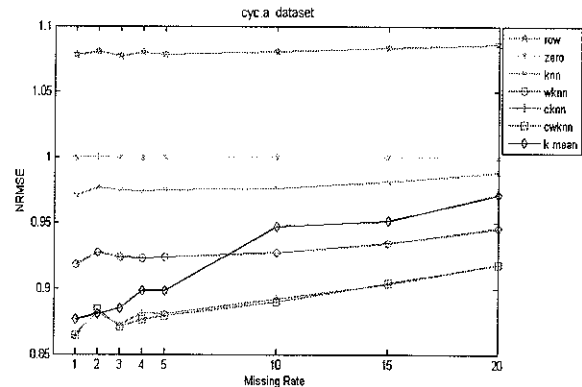


Fig.3. Experimental results on cyc.a dataset, NRMSE measures of investigated methods at different missing rates.

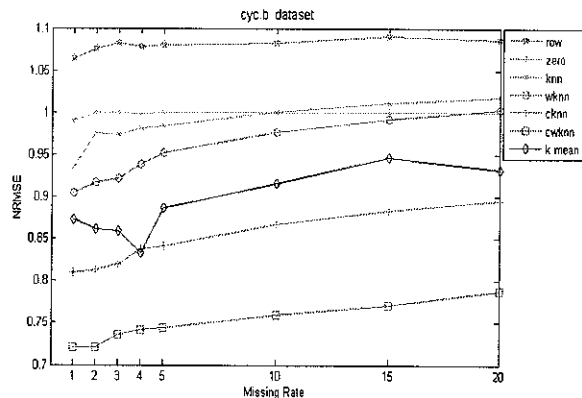


Fig.4. Experimental results on cyc.b dataset, NRMSE measures of investigated methods at different missing rates.

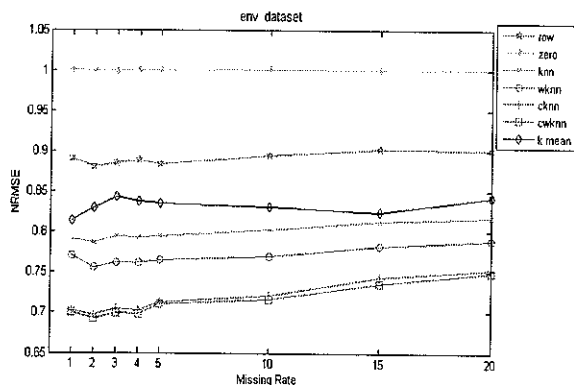


Fig.5. Experimental results on env dataset, NRMSE measures of investigated methods at different missing rates.

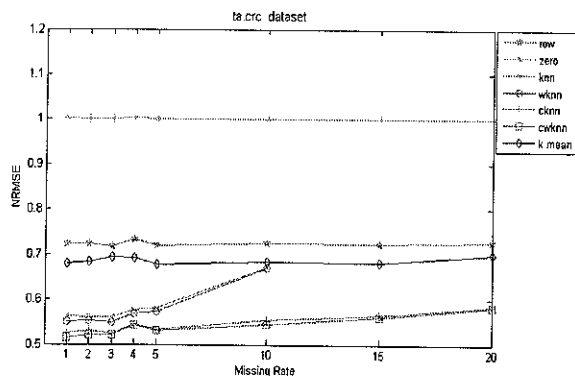


Fig.6. Experimental results on tar-crc dataset, NRMSE measures of investigated methods at different missing rates.

REFERENCES

- [1] T. Aittokallio. Dealing with missing values in large-scale studies: microarray data imputation and beyond. *Briefings in Bioinformatics*, 11(2):253–264, 2010.
- [2] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpen, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, and V. Sondak. Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, 406(6795):536–540, 2000.
- [3] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(4):224–227, 1979.
- [4] A.G. de Brevem, S. Hazout, and A. Malpertuy. Influence of microarrays experiments missing value on the stability of gene groups by hierarchical clustering. *BMC Bioinformatics*, 5:114, 2004.
- [5] A. P. Gasch, M. Huang, S. Metzner, D. Botstein, S. J. Elledge, and P. O. Brown. Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog Mec1p. *Mol. Biol. Cell.*, 12(10):2987–3003, 2001.
- [6] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.
- [7] N. Iam-on, T. Boongoen, and S. Garrett. LCE: A link-based cluster ensemble method for improved gene expression data analysis. *Bioinformatics*, 26(12):1513–1519, 2010.
- [8] N. Iam-on and S. Garrett. LinkCluE: A MATLAB package for link-based cluster ensembles. *Journal of Statistical Software*, 36(9):1–36, 2010.
- [9] I. T. Jolliffe. *Principal Component Analysis*. Springer, NY, 2002.
- [10] R. Jomsten, H. Y. Wang, and W. J. Welsh. DNA microarray data imputation and significance analysis of differential expression. *Bioinformatics*, 21:4155–4161, 2005.
- [11] H. Kim, G. H. Golub, and H. Park. Missing value estimation for DNA microarray gene expression data: Local least squares imputation. *Bioinformatics*, 20:1–12, 2005.
- [12] W. Ling and F. Dong-Mei. Estimation of missing values using a weighted k-nearest neighbors algorithm. In *Proceedings of International Conference on Environmental Science and Information Application Technology*, pages 660–663, 2009.
- [13] S. Oba, M. A. Sato, and I. Takemasa. A bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19:2088–2096, 2003.
- [14] M. S. B. Sehgal, L. Gondal, and L. S. Dooley. Collateral missing value imputation: a new robust missing value estimation algorithm for microarray data. *Bioinformatics*, 21(10):2417–2423, 2005.
- [15] R. Spang. Diagnostic signatures from microarrays: a bioinformatics concept for personalized medicine. *BIOSILICO*, 1:264–268, 2003.
- [16] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell.*, 9:3273–3297, 1998.
- [17] I. Takemasa, H. Higuchi, H. Yamamoto, M. Sekimoto, N. Tomita, S. Nakamori, R. Matoba, M. Monden, and K. Matsubara. Construction of preferential cDNA microarray pecialized for human colorectal carcinoma: molecular sketch of colorectal cancer. *Biophys. Res. Commun.*, 285:1244–1249, 2001.
- [18] O. Troyanskaya, M. Cantor, and G. Sherlock. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17:520–525, 2001.
- [19] J. Tuikkala, L. L. Elo, and O. S. Nevalainen. Missing value imputation improves clustering and interpretation of gene expression microarray data. *BMC Bioinformatics*, 9:202, 2008.
- [20] A. Wallqvist, A. Rabow, R. Shoemaker, E. Sausville, and D. Covell. Establishing connections between microarray expression data and chemotherapeutic cancer pharmacology. *Mol Cancer Ther*, 1:311–320, 2002.
- [21] D. Wang, Y. Lv, and Z. Guo. Effects of replacing the unreliable cDNA microarray measurements on the disease classification based on gene expression profiles and functional modules. *Bioinformatics*, 22:2883–2889, 2006.
- [22] S. Zhang, Z. Qin, C. X. Ling, and S. Sheng. Missing is useful: Missing value in cost-sensitive decision trees. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1689–1693, 2005.
- [23] S. Zhang, C. Zhang, and Q. Yang. Information enhancement for data mining. *IEEE Intelligent Systems*, 19(2):12–13, 2004.