

An Improvement of Missing Value Imputation in DNA Microarray Data Using Cluster-based LLS Method

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

Tossapon Boongoen
Department of Mathematics and Computer Science
Royal Thai Air Force Academy
Bangkok, Thailand
email: tossapon_b@rtaf.mi.th

Abstract—Gene expressions measured during a microarray experiment usually encounter the native problem of missing values. These are due to possible errors occurring in the primary experiments, image acquisition and interpretation processes. Leaving this unsolved may critically degrade the reliability of any consequent downstream analysis or medical application. Yet, a further study of microarray data may not be possible with many standard analysis methods that require a complete data set. This paper introduces a new method to impute missing values in microarray data. The proposed algorithm, CLLS impute, is an extension of local least squares 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 corresponds to genes and experiments, respectively. CLLS kicks off by finding a complete dataset via the removal of rows with missing value(s). Then, gene 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 to the target. Having known this, the target gene is imputed by applying regression analysis with similar genes previously determined. Empirical evaluation with several published gene expression datasets suggest that the proposed technique performs better than the classical local least square method and recently developed techniques found in the literature.

Keywords—missing value, imputation, microarray data, clustering, regression

I. INTRODUCTION

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 [19], identification of unknown effects of a specific therapy [14], and the creation of classification systems for improved cancer diagnosis [15]. Traditional data mining algorithms such as k -means and agglomerative hierarchical clusterings have proven useful for identifying biologically relevant clusters of tissue samples/experimental conditions and genes [7]. See [6] for an empirical study of advanced clustering algorithms for microarray data analysis.

Despite such benefits, gene expression data are frequently peppered with missing values that may occur for a variety of reasons. These include insufficient resolution, image corruption, array fabrication error, and experimental error during the laboratory processing [12]. 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 [5] the need for repeating experiments, which can be expensive and time consuming. Note also that the repeat of experiments may not guarantee data completeness.

Missing values is an actual yet challenging issue confronted in 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 [20], [21]. In fact, general analysis methods, such as principle component analysis [9], multivariate supervised classification or clustering analysis, require a complete data matrix. 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. 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 neighbour impute (KNN) [18], weighted KNN impute [10], local least squares impute (LLS) [12] and Bayesian principal-component analysis (BPCA) [13], have been introduced in the literature, especially for microarray data (see [1] for a survey). In particular to KNN impute, it searches for k other genes with expressions most similar to that of target gene (with missing values). Then, the missing value is estimated by the average or weighted average of corresponding values recorded for those k closest genes.

Despite its simplicity, the acquisition of nearest neighbours is driven only by proximity. This is effective when neighbours

are spatially distributed, i.e., groups/clusters of neighbours do not exist. Such an assumption and the consequent selection of neighbours can be sub-optimal. This may lead to an inaccurate estimation, especially for those target genes situate by the edge of a cluster. The problem can be resolved by bringing information regarding data clusters into the imputation process, which is the basis of cluster-based KNN impute (CKNN) [11]. Given such motivation, this paper introduces the improved LLS impute, called cluster-based LLS (CLLS). The aforementioned cluster-oriented mechanism is used to determine local neighbours for regression analysis, from which impute values are later derived. 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 CLLS 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. MISSING VALUE IMPUTATION FOR MICROARRAY DATA

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. However, the resulting mass of data, often as millions of measurements, can be overwhelming for human comprehension and pose great challenges on data pre-processing 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 image resolution, dust or scratches on the slide or experimental error during the laboratory process. Generally, missing of data around 1-10% would affect up to 95% of the genes in any microarray experiment [3]. Of course, one possible solution to recover the dataset with missing values is to repeat the experiment, which is inefficient and sometimes infeasible. Henceforth, the missing values in the original data need to be systematically predicted.

Let microarray data be formulated as a gene expression matrix $G \in \mathbb{R}^{m \times n}$ with m rows and n columns corresponding to genes and experiments, respectively. In the matrix G , a row $x_g \in \mathbb{R}^{1 \times n}$ represents expressions of the g -th gene in n experiments. As one of standard imputation techniques, KNN impute [18] 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. In particular, 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. To impute a missing value x_{gi} of gene $x_g, g \in \{1, \dots, m\}$ and experiment $i \in \{1, \dots, n\}$, KNN impute finds 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 , 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:

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

here N_g is the set of k nearest neighbours 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 genes. As for example, the estimated value $\tilde{x}_{gi}$ of the missing x_{gi} can be found from the following.

$$\tilde{x}_{gi} = \sum_{x_\alpha \in N_g} x_{\alpha i} w_\alpha, \quad (3)$$

given that

$$w_\alpha = \frac{1/d_\alpha}{\sum_{j=1}^k 1/d_j}, \quad (4)$$

here k is the number of closest genes and d_j denotes the distance between the j -th gene and the target gene. Similar to KNN approach, LLS impute [12] firstly selected k nearest neighbouring genes of the gene under examination, x_g . Following that, these genes are used for the regression analysis, whose result determines the estimate of a missing value. Recently, a new gene selection framework for KNN impute has been formulated [11], where clusters of genes are used to reduce the scope of nearest neighbour search. The resulting method that is called cluster-based KNN impute (CKNN), has proven more accurate than the classical KNN impute. Note that the weighted variation of this model, i.e., CWKNN, is also introduced as the coupling of the aforementioned gene selection and the weighted KNN impute model.

In addition to these neighbour-guided methods, SVD impute [18] has also been introduced for the preparation of microarray data. It computes SVD of the matrix G' that is obtained by substituting all missing values in the original matrix G with zero or row average. Then, for each target gene x_g , the estimate of missing value can be achieved by regressing x_g against most significant eigengenes. In the study of [13], Bayesian principal-component analysis (BPCA), the imputation of missing value is designed as the simultaneous estimation of probabilistic

model and latent variables within the framework of Bayesian inference.

III. CLUSTER-BASED LLS IMPUTATION

Following the success of CKNN and CWKNN models, this section introduces a new imputation method that combines the cluster-guided gene selection and LLS impute. This so-called cluster-based LLS impute includes two majors steps of (i) finding a number of clusters and generating a clustering model that will be used as a reference for the gene selection, and (ii) applying the LLS impute in conjunction with the discovered clusters. These phases are described in the following sections.

A. Acquiring Gene Clusters

Clustering algorithms are 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. The k -means technique 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 matrix $G \in \mathbb{R}^{m \times n}$, 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:

$$J(U, Z) = \sum_{l=1}^k \sum_{i=1}^m \sum_{j=1}^n u_{il} (x_{ij} - z_{lj})^2, \quad (5)$$

where $\sum_{l=1}^k u_{il} = 1$, $U \in \mathbb{R}^{m \times k}$ 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., $z_l = (z_{l1}, \dots, z_{ln}), \forall l = 1 \dots k$.

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 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. 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 higher computational time requirement.
- 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 Davies-Bouldin (DB) measure [2] is taken as the optimal number of clusters for the next processing stage. Note that DB is one of standard validity indices that are normally used to assess the goodness of clustering results. This measure of a partition π can be defined by the following.

$$DB(\pi) = \frac{1}{k} \sum_{C_a, C_b \in \pi} \frac{\delta_a + \delta_b}{d(z_a, z_b)}, \quad (6)$$

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.

B. Applying Cluster-Based LLS Imputation

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, it 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_s only when

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

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 (or experiments). For this purpose, the row average is applied to make a reasonable guess.

Following that, the cluster membership previously discovered is utilized to determine the gene set $N_{x_g^*}$ for a particular target gene x_g^* . The size of this gene set may be specified subjectively by a human expert, but a data-driven counterpart is usually preferred for better adaptability. Specific to this study, the number of genes in $N_{x_g^*}$ is dynamically determined by the intuition that $x_g \in N_{x_g^*}$ only when

$$d(x_g, x_g^*) < \mu^*, \forall x_g \in C_s \quad (8)$$

$$\mu^* = \frac{\sum_{x_g \in C_*} d(x_g, x_g^*)}{|C_*|}, \quad (9)$$

where the target gene x_g^* belongs to cluster C_* , and $|C_*|$ denotes the size of C_* (or the number of genes in this cluster). Having achieved $N_{x_g^*}$, the missing value in x_g^* is estimated from the regression formula, which is derived from those genes of $N_{x_g^*}$. Please consult [12] for mathematical details of the underlying least squares problem. It is noteworthy that CLLS has improved the efficiency of the original LLS model, whose time complexity is $O(m^2)$. 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 $q \ll m$.

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 [4]. 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 pre-specified 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:

$$NRMSE = \frac{\sqrt{\text{mean}(x_{\text{guess}} - x_{\text{ans}})^2}}{\text{std}(x_{\text{ans}})}, \quad (10)$$

where x_{guess} and x_{ans} are imputed values and known answer values presented in the original data matrix, respectively. In

addition, $\text{std}(x_{\text{ans}})$ 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 the examined datasets are shown in Table I. These techniques include the LLS impute that is the baseline of the proposed method, the conventional k nearest neighbors (KNN) and both cluster-based KNN imputation models, i.e., CKNN and CWKNN. To reinforce this evaluation, the by-product of the proposed CLLS is used to form another baseline technique (KM). The centroids of gene clusters acquired during the first phase of CLLS is directly used as the estimate for any target gene. Suppose that cluster C_* is the closest to the target gene x_g^* . The missing dimensional value $x_{g_i}^*$ can be replace by the i -th dimension of centroid z_* (of C_*). The presented NRMSE values are the average of 10 trials, each with the corresponding standard deviation. Note that the results shown in this section are obtained when the size of nearest neighbors used by KNN, CKNN and CWKNN are 10, 5 and 3, respectively. These settings are suggested in the original study of the underlying technique.

According to this table, the cluster-based LLS imputation method (CLLS) perform consistently better than the classical KNN impute. This also holds true as compared to the two cluster-based KNN models, i.e., CKNN and CWKNN. These differences are obvious on Alpha, cyc.a and cyc.b, as compared to the other three datasets. In addition, it is certain that CLLS is generally more effective than the KM baseline over all the six datasets. However, KM achieves reasonable results when the level of missing value increases. As compared to the base line model of LLS, CLLS can improve the accuracy of estimates in all datasets examined herein. This improvement usually gets larger with the amount of missing values. As for example, Figs 1-3 show the focused comparisons of CLLS and LLS on cyc.a, env and ta.crc, respectively.

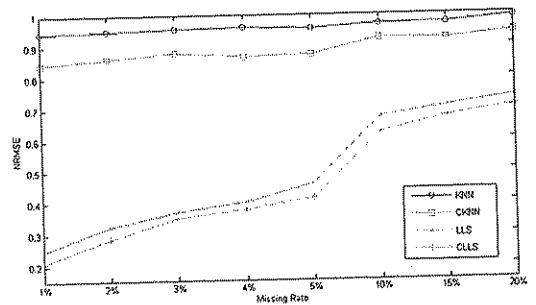


Fig. 1. Focused comparison of NRMSE measures obtained by KNN, CKNN, LLS and CLLS on cyc.a dataset.

V. CONCLUSION

This paper proposes a novel version of local least square (LLS) imputation method to estimate the missing values in

TABLE I
NRMSE MEASURES OF DIFFERENT IMPUTATION METHODS ON THE INVESTIGATED DATASETS. THE CORRESPONDING STANDARD DEVIATION VALUES ARE GIVEN IN PARENTHESES.

Dataset	Missing Data	KNN	CKNN	CWKNN	KM	LLS	CLS
Alpha	1%	0.9444 (0.0158)	0.8886 (0.0204)	0.8442 (0.0198)	0.8177 (0.0161)	0.2271 (0.2231)	0.2088 (0.0315)
	2%	0.9495 (0.0098)	0.8994 (0.0121)	0.8603 (0.0233)	0.8741 (0.0411)	0.3039 (0.0349)	0.2837 (0.0370)
	3%	0.9561 (0.0116)	0.9048 (0.0170)	0.8779 (0.0277)	0.8716 (0.0385)	0.3693 (0.0191)	0.3498 (0.0227)
	4%	0.9592 (0.0107)	0.8949 (0.0166)	0.8651 (0.0233)	0.8714 (0.0200)	0.4005 (0.0124)	0.3765 (0.0121)
	5%	0.9555 (0.0073)	0.9055 (0.0095)	0.8717 (0.0112)	0.8743 (0.0233)	0.4389 (0.0205)	0.4134 (0.0199)
	10%	0.9676 (0.0112)	0.9241 (0.0096)	0.9235 (0.0123)	0.8673 (0.0197)	0.6606 (0.0131)	0.6216 (0.0128)
	15%	0.9712 (0.0092)	0.9292 (0.0060)	0.9203 (0.0080)	0.8781 (0.0145)	0.7042 (0.0079)	0.6709 (0.0057)
	20%	0.9889 (0.0144)	0.9385 (0.0052)	0.9407 (0.0074)	0.8832 (0.0140)	0.7338 (0.0073)	0.7039 (0.0075)
cyc.a	1%	0.9705 (0.0191)	0.9400 (0.0190)	0.8582 (0.0325)	0.8544 (0.0514)	0.2404 (0.0518)	0.1950 (0.0450)
	2%	0.9773 (0.0154)	0.9499 (0.0140)	0.8822 (0.0319)	0.8698 (0.0145)	0.2953 (0.0227)	0.2462 (0.0179)
	3%	0.9751 (0.0153)	0.9460 (0.0110)	0.8722 (0.0254)	0.8617 (0.0392)	0.3730 (0.0515)	0.2868 (0.0234)
	4%	0.9745 (0.0745)	0.9493 (0.0124)	0.8751 (0.0184)	0.8729 (0.0296)	0.4135 (0.0246)	0.3324 (0.0173)
	5%	0.9750 (0.0060)	0.9509 (0.0066)	0.8788 (0.0124)	0.8660 (0.0236)	0.4572 (0.0275)	0.4067 (0.0546)
	10%	0.9767 (0.0071)	0.9546 (0.0045)	0.8947 (0.0173)	0.8664 (0.0159)	0.7639 (0.0204)	0.5736 (0.0155)
	15%	0.9811 (0.0039)	0.9698 (0.0027)	0.9042 (0.0106)	0.8815 (0.0112)	0.8074 (0.0105)	0.6360 (0.0111)
	20%	0.9882 (0.0059)	0.9798 (0.0055)	0.9184 (0.0096)	0.8786 (0.0124)	0.8253 (0.0113)	0.6688 (0.0079)
cyc.b	1%	0.9811 (0.0541)	0.9682 (0.0693)	0.9226 (0.1701)	0.8724 (0.0325)	0.2474 (0.0957)	0.2345 (0.0808)
	2%	0.9760 (0.0403)	0.9586 (0.0550)	0.9045 (0.1281)	0.8547 (0.0854)	0.3972 (0.0774)	0.3663 (0.0690)
	3%	0.9739 (0.0343)	0.9550 (0.0257)	0.9082 (0.0343)	0.8522 (0.0542)	0.4638 (0.0318)	0.4309 (0.0639)
	4%	0.9817 (0.0364)	0.9620 (0.0364)	0.9126 (0.0770)	0.8140 (0.0574)	0.5283 (0.0364)	0.4998 (0.0598)
	5%	0.9842 (0.0324)	0.9791 (0.0324)	0.9642 (0.0726)	0.8931 (0.0903)	0.5952 (0.0549)	0.5676 (0.0361)
	10%	1.0004 (0.0211)	0.9839 (0.0141)	0.9527 (0.0318)	0.8731 (0.0330)	0.6958 (0.0463)	0.6664 (0.0300)
	15%	1.0108 (0.0246)	0.9897 (0.0280)	0.9677 (0.0330)	0.9068 (0.0390)	0.7481 (0.0402)	0.7213 (0.0372)
	20%	1.0173 (0.0216)	0.9929 (0.0128)	0.9873 (0.0274)	0.9034 (0.0332)	0.7839 (0.0243)	0.7577 (0.0256)
elu	1%	0.9654 (0.0073)	0.9325 (0.0156)	0.8407 (0.0344)	0.8568 (0.0355)	0.2459 (0.0400)	0.2261 (0.0391)
	2%	0.9729 (0.0077)	0.9498 (0.0103)	0.8740 (0.0203)	0.8753 (0.0184)	0.2940 (0.0419)	0.2536 (0.0198)
	3%	0.9740 (0.0089)	0.9462 (0.0099)	0.8634 (0.0240)	0.8586 (0.0236)	0.3685 (0.0232)	0.3037 (0.0257)
	4%	0.9724 (0.0041)	0.9440 (0.0050)	0.8678 (0.0153)	0.8638 (0.0177)	0.4133 (0.0199)	0.3408 (0.0119)
	5%	0.9752 (0.9752)	0.9478 (0.0068)	0.8686 (0.0183)	0.8465 (0.0422)	0.4448 (0.0138)	0.3649 (0.0130)
	10%	0.9769 (0.0057)	0.9626 (0.0077)	0.8820 (0.0141)	0.8618 (0.0083)	0.7373 (0.0118)	0.5714 (0.0087)
	15%	0.9797 (0.0057)	0.9664 (0.0039)	0.9050 (0.0141)	0.8717 (0.0100)	0.7864 (0.0084)	0.6303 (0.0082)
	20%	0.9822 (0.0082)	0.9856 (0.0075)	0.9085 (0.0057)	0.8789 (0.0053)	0.8101 (0.0057)	0.6715 (0.0065)
env	1%	0.8014 (0.0284)	0.7889 (0.0271)	0.6993 (0.0421)	0.8151 (0.0476)	0.7643 (0.0230)	0.6293 (0.0457)
	2%	0.7863 (0.0231)	0.7709 (0.0223)	0.6926 (0.0341)	0.8011 (0.0409)	0.7778 (0.0287)	0.6117 (0.0396)
	3%	0.7943 (0.0176)	0.7795 (0.0161)	0.6999 (0.0217)	0.8010 (0.0499)	0.7962 (0.0205)	0.6161 (0.0241)
	4%	0.7937 (0.0058)	0.7802 (0.0059)	0.6983 (0.0142)	0.8117 (0.0316)	0.8011 (0.0134)	0.6197 (0.0110)
	5%	0.7942 (0.0156)	0.7826 (0.0148)	0.7109 (0.0196)	0.8040 (0.0340)	0.8077 (0.0232)	0.6256 (0.0210)
	10%	0.8023 (0.0112)	0.7878 (0.0109)	0.7169 (0.0106)	0.8325 (0.0221)	0.7732 (0.0172)	0.6227 (0.0104)
	15%	0.8126 (0.0081)	0.7982 (0.0081)	0.7425 (0.0095)	0.8425 (0.0222)	0.7778 (0.0137)	0.6443 (0.0077)
	20%	0.8168 (0.0092)	0.8184 (0.0068)	0.7496 (0.0064)	0.8499 (0.0081)	0.7749 (0.0089)	0.6510 (0.0060)
ia.crc	1%	0.5396 (0.0542)	0.5230 (0.0436)	0.5199 (0.0383)	0.7386 (0.0533)	0.5967 (0.0644)	0.4889 (0.0432)
	2%	0.5610 (0.0165)	0.5318 (0.0151)	0.5304 (0.0154)	0.7311 (0.0434)	0.5883 (0.0438)	0.4792 (0.0305)
	3%	0.5600 (0.0203)	0.5255 (0.0245)	0.5208 (0.0270)	0.7590 (0.0416)	0.5836 (0.0312)	0.4775 (0.0298)
	4%	0.5769 (0.0136)	0.5427 (0.0127)	0.5433 (0.0148)	0.7591 (0.0378)	0.5905 (0.0146)	0.4695 (0.0204)
	5%	0.5794 (0.0156)	0.5327 (0.0176)	0.5367 (0.0209)	0.7544 (0.0395)	0.6130 (0.0213)	0.4952 (0.0165)
	10%	0.6722 (0.0147)	0.5523 (0.0124)	0.5553 (0.0112)	0.7715 (0.0223)	0.6757 (0.0179)	0.5144 (0.0178)
	15%	NaN NaN	0.5540 (0.0089)	0.5649 (0.0097)	0.7405 (0.0381)	0.7487 (0.0200)	0.5452 (0.0142)
	20%	NaN NaN	0.5698 (0.0075)	0.5855 (0.0065)	0.7569 (0.0427)	0.9201 (0.0196)	0.5737 (0.0078)

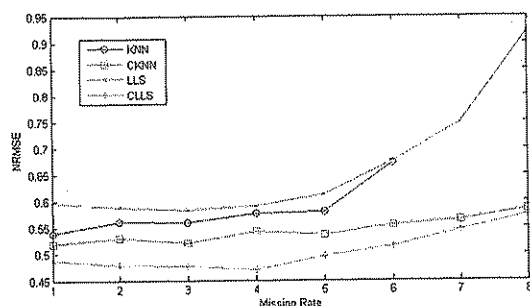


Fig. 2. Focused comparison of NRMSE measures obtained by KNN,CKNN,LLS and CLLS on ta.crc dataset.

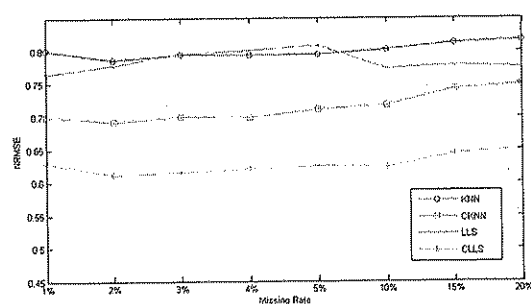


Fig. 3. Focused comparison of NRMSE measures obtained by KNN,CKNN,LLS and CLLS on env dataset.

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 LLS imputation algorithm based on k -means method outperforms the classical LLS impute method, which has been identified as one of the most effective approaches for missing value imputation. The empirical study also shows that the overall performance of the LLS based on clustering method is better than the basic LLS method, especially when the percentage of missing values is high. An interesting future work is find a possible weighting scheme to further improve the model introduced herein. In addition, the use of cluster ensembles [8] may help to stabilize the search for gene clusters, which is currently uncertain and different from one run to another.

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] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(4):224–227, 1979.
- [3] A.G. de Brevern, 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.
- [4] 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.
- [5] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.
- [6] N. Iam-On and T. Boongoen. A new locally weighted k-means for cancer-aided microarray data analysis. *Journal of Medical Systems*, 36(1):43–49, 2013.
- [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, T. Boongoen, S. Garrett, and C. Price. A link-based approach to the cluster ensemble problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2396–2409, 2011.
- [9] I. T. Jolliffe. *Principal Component Analysis*. Springer, NY, 2002.
- [10] R. Jorntsen, H. Y. Wang, and W. J. Welsh. DNA microarray data imputation and significance analysis of differential expression. *Bioinformatics*, 21:4155–4161, 2005.
- [11] P. Keerin, W. Kurutach, and T. Boongoen. Cluster-based KNN missing value imputation for DNA microarray data. In *Proceedings of IEEE International Conference on System, Man and Cybernetics*, pages 445–450, 2012.
- [12] 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.
- [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] C. M. Perou, T. Sorlie, and M. B. Eisen. Molecular portraits of human breast tumors. *Nature*, 406:747–752, 2000.
- [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] A. Wallqvist, A. Rabow, R. Shoemaker, E. Sausville, and D. Coveil. Establishing connections between microarray expression data and chemotherapeutic cancer pharmacology. *Mol Cancer Ther.*, 1:311–320, 2002.
- [20] 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.
- [21] S. Zhang, C. Zhang, and Q. Yang. Information enhancement for data mining. *IEEE Intelligent Systems*, 19(2):12–13, 2004.