A NOVEL DISTANCE BASED MODIFIED K-MEANS CLUSTERING ALGORITHM FOR ESTIMATION OF MISSING VALUES IN MICRO-ARRAY GENE EXPRESSION DATA

Chandra Das¹, Shilpi Bose², Matangini Chattopadhyay³, Samiran Chattopadhyay⁴

¹,²Department of Computer Sc & Engg., Netaji Subhash Engg. College, Garia, Kolkata, West Bengal, India
³School of Education Technology, Jadavpur University, Kolkata, West Bengal, India
⁴Department of Information Technology, Jadavpur University, Kolkata, West Bengal, India

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 DNA microarray analysis, coexpressed genes provide similar biological functions. The coexpressed genes are those whose expression levels may rise and fall synchronously in response to a set of experimental conditions. Although the magnitude of their expression levels may not be close, the patterns they exhibit can be very much alike/correlated. In this paper, a new distance is proposed to find closest coexpressed genes in an efficient way. Based on this distance a modified K-means clustering algorithm is proposed to accurately predict missing values in microarray gene expression data. The estimation accuracy of the proposed clustering method is compared with the widely used KNNimpute, SKNNimpute and IKNNimpute methods on various microarray data sets with different rate of missing entries. The experimental results show the effectiveness of this proposed method compared to other existing methods in terms of Normalized Root Mean Square error.

Keywords: Microarray, Clustering, K-Means, Missing Value Estimation, Coexpressed Gene, Coregulated Gene.
1. INTRODUCTION

Due to recent advancements and wide use of high-throughput technology, the society is experiencing an explosion of biological data. Meaningful interpretation of this large volume of biological data is increasingly becoming difficult. Consequently, researchers, practitioners, and entrepreneurs from diverse fields are trying to develop sophisticated techniques to store, analyze, and interpret biological data for knowledge extraction; a new field called bioinformatics is evolved. Bioinformatics means conceptualizing biology in terms of molecules and applying informatics techniques to understand and organize information about these molecules on a large scale.

Recently, microarray technology has attained a central role in biological and biomedical research. It is a high-throughput technology that allows monitoring transcription levels of many thousands of genes in particular cells or tissues, giving a global view of gene expression [1,2]. Moreover, it also helps in the study of many biological processes varying from human tumors to yeast sporulation.

Microarray data analysis has been successfully applied to a number of studies over a broad range of biological disciplines including cancer classification by class discovery and prediction [3], identification of unknown effects of a specific therapy [4], identification of genes relevant to a certain diagnosis or therapy [5], cancer prognosis [6] etc.

Data from microarray experiments are usually in the form of large matrices, in which each row corresponds to a single gene, each column corresponds to a sample or condition or experiment, and each entry denotes an expression level of a gene under a specific sample or condition or experiment.

Frequently, these matrices contain a huge amount of missing values (MVs). This is due to occurrences of imperfections during the microarray experiments (e.g. insufficient resolution, spotting problems, and deposition of dust or scratches on slides, blemishes on chip, hybridization error, and image corruption) that create suspected values, which are usually discarded and set as missing [7]. In large scale studies involving tens to thousands 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 [8] are affected. Presence of missing gene expression values constitutes a problem for downstream data analysis [3, 9, 10] since many of the analysis methods employed (e.g. classification, data dimension reduction technique, and model based clustering technique) 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. Moreover, analysis results can also be influenced by the estimates replacing the missing values. Thus, in order to minimize the effect of missing values on analysis and to avoid improper analysis, missing value estimation is an important preprocessing step.

There are several simple ways to deal with missing values. One of the naive approach is to simply ignore the samples containing missing values but this is inappropriate. The other simple approaches include removing the genes with missing values before the analysis (case deletion), or replacing the missing values of a gene with the average of the observed values of that gene (mean substitution) [11]. Another common approach is to replace missing log2 transformed gene expression ratios by zeros [7]. These approaches have disadvantages. Case deletion procedures may bias results if remaining cases are unrepresentative of the entire gene space [12]. Both mean and zero substitutions distort relationships among samples and artificially reduce the variance of samples in the gene [11].
The most classical clustering based method used to estimate missing values employs \( K \)-nearest neighbors algorithm (\( K \)NNimpute) [13]. In this approach, the estimated value is computed from the \( K \) closest expression profiles among the dataset. This approach was applied to DNA chips by Troyanskaya and collaborators to estimate MVs for microarray data and rapidly became one of the most popular methods. Since this pioneer study, more sophisticated clustering based approaches have been proposed such as Sequential \( K \)NN (SKNNimpute) [14], Iterative \( K \)NN (IKNNimpute) [15] etc.

Several statistical and numerical analysis based methods have also been proposed. Among them most popular are approaches based on the Expectation Maximisation (EM) named EM gene and EM array [16], singular value decomposition (SVDimpute) [13] etc. Principle of least square has widely been used to produce different versions of imputation techniques named Ordinary Least Square Imputation (OLSimpute), LSI gene, LSI array, LSI combined, LSI adaptive [16], local least square imputation (LLSimpute) [17]. Bar-Joseph et al. described a model-based spline fitting method for time-series data [18] and Schliep et al. used hidden Markov models for imputation [19]. Bayesian Principal Component Analysis (BPCA) [20] combines a principal component regression, a Bayesian estimation and a variation of Bayes (VB) algorithm to estimate missing values in microarray gene expression data. In addition, the following relevant methodologies are also applied in replacement of MVs for microarray analysis: Support Vector Regression [21], Factor Analysis Regression [21], Gaussian mixture clustering [21], LinCmb [21], Collateral missing value estimation (CMVE) [21], Linear based model imputation [21], Dynamic Time Warping [21] etc.

However, the prediction error of these methods [21] still impacts the performance of statistical and machine learning algorithms such as class prediction, class discovery and differential gene identification algorithms [9]. Thus, there is considerable potential to develop new techniques that may provide minimal prediction errors for different types of microarray data including both time series and non-time series sequences.

To understand gene functionality, gene regulation, cellular processes, and subtypes of cells, clustering techniques [14, 22] have proven to be helpful in microarray gene expression data. However, \( K \)NN, SKNN, and IKNN are clustering based approaches but still now clustering based approaches have not been yet highlighted more [23] to predict missing values in gene expression data. In this paper, emphasis is given on clustering based approach to predict missing values in microarray gene expression data. In this regard, a novel distance based modified \( K \)-means clustering algorithm is proposed here to predict missing values in microarray gene expression data. The proposed method is compared with several existing well-established techniques, namely, \( K \)NNimpute, SKNNimpute, IKNNimpute. Performance of these methods are rigorously evaluated for prediction of randomly introduced missing values in different microarray datasets. The methods are evaluated by comparing their estimates for the artificial missing entries with the true values. Normalized root mean squared error (NRMSE) is used here to quantitatively evaluate the estimation accuracy of these techniques. The effectiveness of the proposed method is demonstrated on three microarray gene expression datasets including both time series and non time series.

2. PROPOSED METHOD

Cluster analysis is a technique [14], which partitions the given data set into several distinct groups for finding natural groups present in the data set. Intuitively, objects in a cluster are as similar as possible and the objects from different clusters are as dissimilar as possible. To understand gene function, gene regulation, cellular processes, and subtypes of
cells, clustering techniques are very much helpful in microarray gene expression data. Most clustering models, define similarity among different objects by distances over either all or only a subset of the dimensions. In other words, similar objects are required to have close values on at least a set of dimensions. Some well-known distance functions include Euclidean distance, Manhattan distance, and cosine distance.

Current research demonstrates that coexpressed genes may have similar cellular functions and so they can be belong to same group or cluster. In DNA microarray analysis, the coexpressed genes are those whose expression levels may rise and fall synchronously in response to a set of environmental stimuli. Although the magnitude of their expression levels may be or may not be close, the patterns they exhibit can be very much alike or correlated. Furthermore, coexpressed genes in the same cluster are likely to be involved in the same cellular processes and a strong correlation of expression patterns between those genes indicates coregulation. If the correlation among genes is exploited properly then missing value prediction error can be reduced significantly [13] in gene expression data.

In this regard a novel distance based $K$-means clustering algorithm is proposed here to predict missing values in microarray gene expression data. The proposed distance is named as hybrid distance which takes advantages of both Euclidean distance and Pearson correlation coefficient in finding coexpressed genes. Using this hybrid distance proposed modified $K$-means algorithm can predict missing values in an efficient way. In the next subsection the proposed distance function is described first and then using this distance function the modified $K$-means clustering algorithm is elaborated.

### 2.1 Proposed new distance measure

In any model, similarity among different objects is defined by distances over all dimensions. In other words, similar objects are required to have close values on at least a set of dimensions. But in case of gene expression data, it is assumed that genes with similar expression patterns (magnitude of their expression levels may be or may not be close) under various experimental conditions or time-course may imply coregulation or relation in functional pathways. So, some well-known distance functions include Euclidean distance, Manhattan distance, Cosine distance functions [24] etc. are not always adequate in capturing pattern based similarity among the genes in gene expression data.

Most popular distance is Euclidean distance. The **Euclidean distance** between points $p$ and $q$ is the length of the line segment connecting them $(pq)$. In Cartesian coordinates, if $p = (p_1, p_2, \ldots, p_n)$ and $q = (q_1, q_2, \ldots, q_n)$ are two points in Euclidean $n$-space, then the distance $(d)$ from $p$ to $q$, or from $q$ to $p$ is given by:

$$d(p, q) = d(q, p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \quad (1)$$

But this type of distance is not suitable to find similarity between genes in gene expression data. As for example the genes considered as similar by Euclidean distance may be very dissimilar in terms of their patterns or vice versa. In fact, if strong pattern similarity/correlation exists between two genes but they are far apart from each other or only
differ from each other by a large multiplication factor then their Euclidean distance is large enough. This is demonstrated with different examples in Figure 1, Figure 2, and Figure 3.

Figure 1 shows a data set with three genes with five samples (attributes). It is easy to see that the all three genes manifest similar patterns on these attributes. However, these genes may not be considered in a cluster by any traditional distance functions including Euclidean distance, Manhattan distance, and cosine distance as the distance between any two of them is not close.

![Figure 1: Shifting patterns in gene expression data](image1)

In figure 2, gene1 is negatively correlated with gene2 and gene3. Although they are highly pattern based similar in opposite direction but Euclidean distances between gene1 and gene2, gene1 and gene3 are large enough.

![Figure 2: Oppositely correlated genes](image2)

![Figure 3: Genes which are closest but pattern based dissimilar](image3)

In figure 3, the Euclidean distance between two genes are very small but they are not pattern based similar.

The Pearson correlation coefficient based similarity is used to overcome the above mentioned issues in gene expression data. It [24] studies the coherence among a set of objects. Pearson correlation coefficient \( P(o_1, o_2) \) defines the correlation between two objects \( o_1 \) and \( o_2 \) as:

\[
P(o_1, o_2) = \frac{\sum (o_1 - \bar{o}_1)(o_2 - \bar{o}_2)}{\sqrt{\sum (o_1 - \bar{o}_1)^2 \times (o_2 - \bar{o}_2)^2}}
\]  

(2)
$\bar{o}_1$ and $\bar{o}_2$ are the mean of all attribute values in $o_1$ and $o_2$, respectively. From this formula, it can be observed that the Pearson correlation measures the correlation between two objects with respect to all attribute values. A large positive value indicates a strong positive correlation while a large negative value indicates a strong negative correlation.

For missing value estimation, if closest coexpressed genes that means expression value based closest and pattern based similar genes (positively correlated or negatively correlated) of target gene are fetched then missing value of the target gene will be predicted more accurately. In this regard, a new hybrid distance is introduced here which takes advantages of both Euclidean distance and Pearson correlation coefficient.

The proposed hybrid distance can be defined as:

$$
H(g_i, g_j) = \begin{cases} 
(1 - |P(g_i, g_j)|) \times d(g_i, g_j) & \text{if } P(g_i, g_j) > 0 \\
(1 - |P(g_i, g_j)|) \times d(g_i, -g_j) & \text{if } P(g_i, g_j) < 0
\end{cases}
$$

Hybrid distance introduced here is a combination of Euclidean distance and Pearson correlation coefficient. It is calculated in the following way. First the similarity between genes $g_i$ and $g_j$ are calculated using Pearson correlation coefficient using equation(2). Then the absolute value of this Pearson correlation coefficient is subtracted from 1 which gives us correlation dissimilarity between two genes whether the two genes are positively correlated or negatively correlated. If correlation coefficient is positive that means two genes are positively correlated then their Euclidean distance is calculated according to eqn(1). But if two genes are negatively or oppositely correlated then sign of any one of two genes let $g_j$ is changed and then Euclidean distance between them is calculated according to eqn(1). Here changing the sign of $g_j$ means $g_i$ and $g_j$ are converted in the same direction of correlation and then their distance is calculated. If $H(g_i, g_j)$ is low, then the genes $g_i$ and $g_j$ are coexpressed and correlated (either positively or negatively). So, using this hybrid distance function the most closest positively correlated or negatively correlated genes can be fetched to predict more accurately missing value in the target gene.

In the next section, first traditional $K$-means clustering algorithm then modified $K$-means clustering algorithm are elaborated.

### 2.2 K-Means Clustering Algorithm

The algorithm proceeds by partitioning $N$ number of objects into $K$ 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

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

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

**Input**: A gene expression matrix $X$ containing $N$ number of genes and $Z$ number of samples. $K$ represents the number of clusters.

**Output**: A set of $K$ clusters.

1) Arbitrarily choose $K$ number of object from $X$ and they are assigned in $m_i$, $i = 1$ to $K$ as initial cluster means or centroids.

2) **Repeat**
3) (Re)Assign each data object \( x_k \) to the cluster \( U_i \) to which the object is the most similar based on the mean value of the objects in the cluster.

4) Update the cluster means, i.e., calculate the mean value of the objects for each cluster using the following equation.

\[
m_i = \frac{\Sigma_{x_k \in U_i} x_k}{|U_i|}
\]  

(5)

Where \(|U_i|\) represents number of objects in cluster \( U_i \).

5) Until criterion function converges, i.e., there are no more new assignments.

The modified-means clustering algorithm is elaborated next in which centroids are calculated in an efficient ways.

2.3 Modified K-Means Clustering Algorithm

Input: A gene expression matrix \( K \) containing \( u \) number of genes and \( v \) number of samples. \( K \) represents the number of clusters.

Output: A set of \( K \) clusters.

1) Arbitrarily choose \( K \) number of object from \( X \) and they are assigned in \( m_i, i = 1 \) to \( K \) as initial cluster means or centroids.

2) Repeat

3) (Re)Assign each data object \( x_k \) to the cluster \( U_i \) to which the object is the most similar based on the mean value of the objects in the cluster.

4) Compute new mean for each cluster using the way given below:

   i) First in every cluster \( U_i \), the most densed object is found that means the object (let \( p \)) whose average Hybrid distance from all other data objects belong to that cluster is minimum and \( m_i = p \).

   ii) For each object \( x_k \) in cluster \( U_i \), the Pearson correlation based similarity between \( x_k \) and \( p \) is calculated.

      a) If similarity > 0 then

      \[
m_i = m_i + x_k
\]

      b) If similarity <0 then

      \[
m_i = m_i - x_k
\]

   iii) Finally, \( m_i = \frac{m_i}{|U_i|} \)

where \(|U_i|\) is the number of objects in cluster \( U_i \).

5) Until criterion function converges, i.e., there are no more new assignments.

2.4 Imputation of missing values

Initially, all missing values in \( X \) are replaced by the estimation given by row (gene) averages to obtain a complete matrix. Specially, this step of gene average substitution, provides the possibility of contributing the maximum number of genes for estimating the missing values. Then the above mentioned clustering algorithm is executed on this complete matrix. The missing values are imputed by making use of the signed mean of the values of the corresponding attribute of all the objects present in the clusters by the following equation.

\[
x_{t,j,l} = \frac{\sum_{x_i \in U_i} \left( x_{ij} \times \frac{p(x_t,x_l)}{p(x_t,x_j)} \right)}{|U_i|}
\]  

(6)
Where $x_{ij}$ is the $j^{th}$ column of target gene $x_t$ (whose missing value is to be imputed) present in the cluster $U_i$ and $x_i$ represents any gene in $U_i$ except $x_t$. $p(x_t, x_i)$ represents Pearson correlation coefficient between $x_t$ and $x_i$. $|U_i|$ represents number of genes in cluster $U_i$.

The main steps of the imputation algorithm is as follows:

1) Initially all missing values in $X$ are replaced by the estimation given by row (gene) averages for obtaining a complete matrix.
2) Apply the above mentioned clustering algorithm to cluster genes.
3) Estimate missing value by using eqn.(6) for all missing entries.

3. EXPERIMENTAL SECTION

In this paper, the performance of missing value estimation using $K$-means clustering algorithm with new hybrid distance is compared with $K$-means clustering with Euclidean distance, $K$-means clustering with Pearson correlation coefficient based similarity, $KNN$, $SKNN$, and $IKNN$ on three different microarray gene expression datasets. The datasets are from yeast and fish. These datasets are classified into three categories:

(1) time-series dataset (SP.AFA) (2) mixed (containing both time series and non-time series datasets, here, GAS) (3) non-time series dataset (Tymchuk). Datasets are described below.

**α-factor dataset of Spellman**

The first dataset (SP.AFA) is obtained from $\alpha$ –factor block release that was studied for the identification of cell-cycle regulated genes in yeast Saccharomyces cerevisiae [29]. This time series dataset consists of 6178 number of genes and 18 number of samples. Among them 4304 number of genes have no missing value. So, a complete data matrix of 4304 genes and 18 samples is prepared that does not contain any missing value to asses missing value estimation methods.

**Gasch’s dataset containing environmental changes in yeast**

The fifth dataset (GAS) is a mixed dataset containing both time series and non-time series datas from a study of response to environmental changes in yeast [30]. It contains 6152 number of genes and 173 number of experiments that have both time-course and non time-course data of specific treatments. After deleting missing rows, a whole mixed dataset of 756 genes and 173 experiments is formed.

**Dataset on Atlantic salmon**

Tymchuk is a Atlantic salmon non-time series affymetrix technology based gene expression dataset (GSE19117 dataset on Atlantic salmon [27]). It consists of 5299 number of genes and 34 number of samples. After deleting gene rows with missing elements a complete matrix containing 1617 number of genes and 34 number of samples is generated.

3.1 Missing data set-up

The datasets used in this study are processed in several steps. Firstly, they are log-transformed (base 2) after they are taken from the image (except for the cases where datasets are already available in log$_2$ scale). Secondly, the original gene expression datasets are then
transformed to complete datasets by removing the genes (rows of the dataset) which contain at least one missing value.

Let an original gene expression matrix $G_{\text{initial}}$ with $p$ initial number of genes and $q$ number of experiments (with real MVs) are given. The initial complete gene expression matrix $G_{\text{complete}}$ with $r$ number of genes and $q$ experiments is built by removing genes with missing entries from the matrix $G_{\text{initial}}$. To evaluate the accuracy of an imputation method, artificial missing entries are introduced to the complete gene expression matrix (i.e. without MVs), $G_{\text{complete}}$. That means the test matrix $G$ is constructed by introducing artificial missing entries into $G_{\text{complete}}$. For introducing artificial missing entries in the gene expression matrix, two procedures A and B are considered.

1. **Missing at random (Uniform):** In procedure A, the test set $G$ is constructed by randomly removing (marked as missing) a specific percentage of the entries (1, 5, 10, 15 and 20%) from $G_{\text{complete}}$. These percentages are chosen based on the values of missing rate commonly encountered in real experimental microarray datasets. A simple random function is used as is done in several works in the literature [15].

2. **Missing at random (Non-uniform):** Given that the probes are arrayed at random in the chips, one can expect that the missing signals caused by effects such as irregularities in the spot production, hybridization failure, dust on the chip, spatial noise, etc., will have a non uniform distribution. However, in some cases where the signal is too low, the image processing software used for spotted cDNA microarrays flags out signals that cannot be distinguished from the background, or that have too irregular shapes. In such cases, missing entries are not uniformly present. Instead, position of missing data depends on signal intensity. In general, it should be expected that a combination of uniform and non uniform missing values will be present in a given microarray dataset. Procedure B, intends to reproduce such occurrences of realistic missing values. In procedure B, MVs are introduced to the elements in the randomly selected $l$ rows of the complete matrix. In each selected row, a number of consecutive columns are set as missing. De Brevern et al. [8] examined the content of MVs in eight series of microarray experiments and reported that percentage of MVs varied from 0.8 to 10.6%. It has also been shown that only a small number of genes (inferior to 1.5%) have more than 50% of their entries missing, while there is no array with more than 50% entries missing. Procedure B results in a structure that is similar to the structure due to De Breven. The total percentage of missing elements in our test datasets generated by procedure B is varied between 1.5 to 4.5%.

3.2 Selection of values for model parameters of different methods

   Every clustering method is executed for $K = 5$ to 50, where $K$ is the number of clusters. The experiments show that for $K > 50$ the clustering results deteriorates. For every clustering method best result (i.e. minimum NRMS error) is taken for different values of number of clusters ($K$). The result is shown for different rate of missing entries present in every data set.

   $k$NNimpute, $sk$NNimpute, and $i$kNNimpute 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 NRMS errors of these three methods are shown by varying $K$ between 5 to 20 in every data set with different rates of missing values.
3.3 Assessment of Performance

To analyze the performance of different algorithms, the experimentation is done on three microarray gene expression datasets. The datasets contain time series data, non time series data and their combinations. Missing values are introduced in each data set in different percentages. The positions of missing values are uniformly and non uniformly distributed. The metric used for evaluating the performance of different algorithms is Normalized Root Mean Squared Error (NRMSE).

**Normalized Root Mean Square Error**

For every dataset, each estimation method is applied to predict the artificially introduced missing values, and the accuracy of the method is evaluated by calculating the error between actual \( y_h \) and estimated values \( \hat{y}_h \) in terms of normalized root mean squared error (NRMSE). It is calculated as follows:

\[
NRMSE = \frac{1}{\sigma_y \sqrt{n}} \sum_{h=1}^{n} (y_h - \hat{y}_h)^2
\]  

(7)

Where \( \sigma_y \) is the standard deviation for the \( n \) true values that correspond to all missing entries in the test matrix.

3.4 Comparative Performance Analysis based on NRMS error

The efficiency of the newly proposed hybrid distance based \( K \)-means method is compared with \( K \)-means using Euclidean distance, \( K \)-means using Pearson correlation based similarity, \( KNN \)impute, \( S KNN \)impute, and \( I KNN \)impute methods, by applying them to three different types of microarray datasets with different missing rates. The metric used is NRMS error.

Figure 5, 6, and 7 plot the performance of different methods as a function of various percentages (1, 5, 10, 15, 20% and unequal) of missing entries on different datasets. The performance is judged by the NRMS error value. NRMS Error tends to decrease with increasing percentage of MVs for each method.

![Figure 5: Comparison of NRMS error for Different methods over different rate of missing entries in dataset SP.AFA](image-url)
In Figure 5, in case of SP.AFA dataset, $K$-mean with Pearson correlation coefficient, and SKNNimpute give similar results for different missing rates. Their NRMS error values range mainly from 0.8 to 0.9. For SP.AFA, KN Nimpute, IKNNimpute, $K$-mean with Euclidean distance, and $K$-mean with hybrid distance show similar type of results. Their NRMS error values range mainly from 0.65 to 0.75.

In Figure 6, for Gasch dataset, $K$-mean with Pearson correlation coefficient, and SKNNimpute give similar type of results for different missing rates. Their NRMS error values range mainly from 0.85 to 0.9. For this dataset, KN Nimpute, IKNNimpute, $K$-mean with Euclidean distance, and $K$-mean with hybrid distance show similar type of results. Their NRMS error values range mainly from 0.65 to 0.75.

In Figure 7, in case of Tymchuk dataset, $K$-mean with Pearson correlation coefficient, and SKNNimpute give similar results for different missing rates. Their NRMS error values range mainly from 0.7 to 0.8. For this dataset, KN Nimpute, IKNNimpute, $K$-mean with Euclidean distance, and $K$-mean with hybrid distance show similar type of results. Their NRMS error values range mainly from 0.48 to 0.6.
Considering all the results it can be concluded that the proposed method has given best results compared to all other methods mentioned here for all different rates of missing entries in all datasets.

4. CONCLUSION

In this paper, a new hybrid distance based modified K-mean clustering algorithm based imputation method is proposed for estimating missing values in DNA microarray data. The prediction performance of the proposed method is assessed and compared with other well known existing methods over different types of datasets (time series, mixed and non-time series datasets) with different proportions (1, 5, 10, 15 and 20%) using different values of K( number of nearest neighbouring genes. Using NRMSE, it is found that the proposed method outperforms all other methods mentioned here for different datasets. So, it is a new robust approach to estimate missing values in microarray gene expression data.

5. REFERENCES