Incorporating Nonlinear Relationships in Microarray Missing Value Imputation

Tianwei Yu, Hesen Peng, and Wei Sun

Abstract—Microarray gene expression data often contain missing values. Accurate estimation of the missing values is important for downstream data analyses that require complete data. Nonlinear relationships between gene expression levels have not been well-utilized in missing value imputation. We propose an imputation scheme based on nonlinear dependencies between genes. By simulations based on real microarray data, we show that incorporating nonlinear relationships could improve the accuracy of missing value imputation, both in terms of normalized root-mean-squared error and in terms of the preservation of the list of significant genes in statistical testing. In addition, we studied the impact of artificial dependencies introduced by data normalization on the simulation results. Our results suggest that methods relying on global correlation structures may yield overly optimistic simulation results when the data have been subjected to row (gene)-wise mean removal.

Index Terms—gene expression, statistical analysis, missing value.

1 INTRODUCTION

M ICROARRAY and other high-throughput data often contain missing values that hamper the application of many unsupervised and supervised learning techniques. A number of methods were developed for the imputation of the missing values, some of which were reviewed and compared by Brock et al. [1]. Most imputation methods stem from the observation that the relationship between gene expression levels can be approximated by linear models. There are two broad classes of imputation methods. The first class utilizes local information, i.e., neighbors of the gene of interest. A representative of this class is the widely used K-nearest neighbor (KNN) method, which fills the missing value by the weighted average of the expression of K neighboring genes in the same array [2]. The L.Simpute method applies linear regression between the gene with missing value and neighboring genes [3]. Robust regression using the principal components of the neighbors was also proposed [4]. Local least squares (LLSs) imputation uses multiple regression of the gene with missing value against the neighboring genes [5]. The method often achieves optimal performance with a very large number of neighbors, some times over 50 percent of the genes in the array [1], [5]. This suggests the method partly takes advantage of the global correlation structures between arrays. A multistage clustering method which incorporates missing value imputation was also proposed [6]. The second class of imputation methods utilizes the global correlation structure.

The SVD method uses the dependency of the experiments on the first singular vectors of the data matrix to impute the missing values [2]. A more complex and better performing variant, BPCA, uses probabilistic PCA and Bayesian estimation [7]. To go beyond the linear relationship between experiments, the Support Vector Regression imputation maps the samples to higher dimensional space with kernel functions [8].

In addition to using a single imputation scheme, the idea of ensemble learning has been applied, in which imputed values from multiple methods were combined [9], [10], [11]. Another way of improving imputation accuracy is to recruit external information. Examples include the use of other microarray data sets from the same organism [12], functional annotation of genes [13], knowledge of synchronization loss in time series data [14], and epigenetic information [15].

Dependencies between genes can be complex and far from linear [16], [17]. General dependency has been shown to be effective for the inference of gene regulatory networks [17], [18], [19]. These results suggest that nonlinear relationships between genes are biologically relevant, and may provide extra information in missing value imputation. A parametric nonlinear regression method has been proposed to utilize such information, which however still preselects neighboring genes based on euclidean distance [20]. Here, we present a missing value imputation scheme that follows the nearest neighbor principal. The definition of “neighbors” is broadened to incorporate nonlinear dependency. The method is based on both Pearson’s correlation coefficient and a new metric for detecting nonlinear relationships, named similarity based on conditional ordered list (SCOL). For neighbors selected by SCOL, the points corresponding to the expression levels may not be geometrically close to each other. Consequently, the imputation is done using the principle of estimating nonlinear response curves by kernel smoother between the gene of interest and its neighbors.
2 METHODS

2.1 Similarity Based on Conditional Ordered List

Here, we present a similarity score between two vectors \( \mathbf{x} \) and \( \mathbf{y} \), the purpose of which is to measure the spread of the conditional distribution \( Y|X \) in a nonparametric manner. First, each vector is standardized using normal score transformation. The \( m \) values of the vector are compared to obtain the ranks \( R_1, R_2, \ldots, R_m \), and then each \( x_i \) of the vector is replaced by \( \Phi^{-1}(R_i/(m + 1)) \), where \( \Phi(.) \) is the cumulative normal distribution. Second, we sort the data \( \{(x_i, y_i)\} \) by the \( X \) values: \( \{(x^*_i, y^*_i)\}, x^*_1 \leq x^*_2 \leq \cdots \leq x^*_m \). Third, the sum of absolute differences between the adjacent \( Y \) values in the ordered list is obtained as the raw distance,

\[
D_{\text{col}}(\mathbf{x}, \mathbf{y}) = \sum_{i=2}^{m} |y^*_i - y^*_{i-1}|.
\]

This is equivalent to summing the Manhattan distances between adjacent points in the list ordered by \( X \), up to a constant. Last, the raw distance is standardized such that it is between zero and one. The theoretical minimum of the distance is

\[
D_{\text{col}}^{\text{min}} = \Phi^{-1}(m/(m + 1)) - \Phi^{-1}(1/(m + 1)),
\]

which is obtained when the two vectors have the exact same rank order. When \( m \) is odd, the theoretical maximum possible distance is

\[
D_{\text{col}}^{\text{max}} = 2 \sum_{i} |\Phi^{-1}(i/(m + 1))| + \Phi^{-1}((m - 1)/2(m + 1)).
\]

When \( m \) is even,

\[
D_{\text{col}}^{\text{max}} = 2 \sum_{i} |\Phi^{-1}(i/(m + 1))| + 2\Phi^{-1}(m/2(m + 1)).
\]

These maximum possible values are obtained when all the adjacent \( Y \) values in the ordered list have opposite signs and the two end \( y \) values have the smallest absolute values. To standardize the raw score such that it is between zero and one, we obtain the SCOL similarity by

\[
S_{\text{col}}^{\text{std}}(\mathbf{x}, \mathbf{y}) = 1 - \frac{D_{\text{col}}(\mathbf{x}, \mathbf{y}) - D_{\text{col}}^{\text{min}}}{D_{\text{col}}^{\text{max}} - D_{\text{col}}^{\text{min}}}.
\]

This SCOL similarity is asymmetric, as it focuses on the spread of \( Y|X \) (Figs. 1a & 1b). For the purpose of missing value imputation, we use the asymmetric version \( S_{\text{col}}^{\text{std}}(\mathbf{x}, \mathbf{y}) \) where \( Y \) is the gene with missing value to be imputed and \( X \) is any other gene. To obtain a symmetric version for measuring nonlinear relationship, we can take

\[
S_{\text{col},\text{sym}}^{\text{std}}(\mathbf{x}, \mathbf{y}) = \max\{S_{\text{col}}^{\text{std}}(\mathbf{x}, \mathbf{y}), S_{\text{col}}^{\text{std}}(\mathbf{y}, \mathbf{x})\}.
\]

2.2 Selecting Informative Genes

To impute the missing values of a gene, we utilize both nonlinear dependency (measured by SCOL) and linear dependency (measured by the absolute value of Pearson's correlation coefficient) to select informative genes. The reason for combining CC and SCOL in gene selection is because CC is more responsive when a slight curvature exist, yet the relationship is still monotone, while SCOL is more responsive when the curvature is large and/or the relation is nonmonotone.

First, we compute the absolute correlation matrix \( \mathbf{C} \) and the SCOL matrix \( \mathbf{B} \), in which \( b_{ij} = S_{\text{col}}^{\text{std}}(\mathbf{x}_i, \mathbf{x}_j) \). To compute the \( i \)th row in \( \mathbf{B} \), we reorder the columns of the expression matrix \( \mathbf{G}_{n \times m} \) based on the rank order of the \( i \)th gene to generate \( \mathbf{G}_{i\times m}^{\text{R}} \). Then, absolute differences are taken between adjacent columns of the reordered matrix to generate the

---

Fig. 1. Illustration of the similarity based on conditional ordered list. (a) An example where \( Y \) and \( X \) have a nonlinear relationship and the spread of \( Y|X \) is small. (b) The same data as in (a) with the role of \( X \) and \( Y \) reversed, resulting in much lower similarity score because the spread of \( Y|X \) is large. Red vertical line segments: distance between adjacent \( Y \) values based on the list ordered by the value of \( X \). Blue dotted horizontal line segments: distance between adjacent \( X \) values.
2.3 Imputing the Missing Value

Since nonlinear relations are involved, and each informative gene may show a different response pattern with the gene of interest, we use a procedure based on kernel smoother in conjunction with weighted averaging for the imputation. In steps 1 to 4 below, we find an imputed value from one informative gene. We use \( g_x \) to denote the gene expression profile with a missing value to be imputed, and let \( g_x \) be missing. We use \( g_x \) to denote the expression profile of an informative gene. To simplify the discussion, we assume both expression profiles are nonmissing at positions \( 2, \ldots, p \). Steps 1 to 3 are simply steps of fitting a kernel smoother. Because our interest is in the fit at the missing value location, only a single point of the kernel smoother is calculated.

1. Find the difference between the nonmissing values of the two expression profiles.

\[
d_{ij} = g_{xj} - g_{ij}, \quad j = 2, \ldots, p.
\]

The purpose of using the difference, rather than \( g_x \) itself, is to alleviate underestimation of extreme values when the relation is monotone.

2. Find the weight based on the Gaussian density. Observations with \( g_{xj} \) closer to \( g_{x1} \) receive higher weights.

\[
w_{ij} = e^{-(g_{xj} - g_{x1})^2/2\sigma^2}, \quad j = 2, \ldots, p,
\]

where \( \sigma \) is one half of the 1/3 quantile of \( |g_{xj} - g_{x1}|, \quad j = 2, \ldots, p \). This ensures that 1/3 of the samples are within \( 2\sigma \), and make major contribution to the imputation.

3. Impute the missing value by weighted mean,

\[
\hat{g}_{x1} = \frac{\sum_{j=2}^{p} d_{ij} w_{ij}}{\sum_{j=2}^{p} w_{ij}} + g_{x1}.
\]

4. Take the weighted variance of the \( d \) vector for a measure of confidence,

\[
s_d^2 = \frac{\sum_{j=2}^{p} w_{ij} (d_{ij} - \hat{d}_{i1})^2}{\sum_{j=2}^{p} w_{ij}}.
\]

5. Merge the imputed values from different informative genes by weighted averaging. The weight has two components: 1) genes that have higher SCOL similarity with the gene of interest receive higher weight, and 2) predictions with smaller weighted variance receive higher weight. We assign the weight by

\[
\delta_i = 1/e^{-6S_{col}(g_i, g_1) \delta_i},
\]

The reason the parameter value 6 was chosen for the exponential decay function was that we observed empirically most top neighbors showed similarity scores between 0.5 and 0.9. With the parameter 6, the weight increases roughly 10 fold from an SCOL score of 0.5 to an SCOL score of 0.9. The final imputed value is the weighted average

\[
\hat{g}_{x1} = \frac{\sum_{i=1}^{n} \delta_i \hat{g}_{x1}}{\sum_{i=1}^{n} \delta_i},
\]

where \( n \) is the number of informative genes used.

2.4 Simulation Study

2.4.1 Data Sets

Six data sets were used in the simulation study. They included the B-cell lymphoma profiling data [21], the data set of yeast transcriptome/translatome comparison [22], the NCI60 cell line gene expression data [23], and the GSE19119 data set on Atlantic salmon [24]. Two yeast cell cycle time series [25], the alpha factor data set and the elutriation data set, were used to probe the effect of data normalization on simulation results in imputation studies. The number of genes/conditions of each data set is listed in Tables 1 and 2.

2.4.2 Methods Compared

Four popular imputation methods were used for comparison. They included the KNN method [2], the Bayesian PCA (BPCA) method [7], the LLS method [5], and the SVD method [2]. The R implementations of these methods were used, which involved packages “impute” and “pcaMethods” [27]. In the results, we refer to our method as “NL.” For NL and KNN, the data matrix (after masking known expression values) was first row (gene)-standardized before the imputation, and the values were transformed back after imputation.

2.4.3 Simulating Missing Patterns

Different percentages of missing patterns (1 percent, 5 percent, 10 percent, 15 percent and 20 percent) were simulated. To simulate data with realistic missing patterns, we used a scheme to learn the missing patterns directly from the real data. For every data set, we divided the data into two parts—matrix \( A \) containing all the genes without any missing value, and matrix \( B \) containing genes with missing values. The simulated data were generated from matrix \( A \). Assume matrix \( A \) had \( n \) rows and \( m \) columns, and we targeted to obtain a simulated data matrix with \( p \) percent
missing values. For every gene $g_x$ in matrix $A$, we first took a random sample from the Poisson distribution with $\lambda = mp/100$ to assign the number of missing entries to the gene. Assume the assigned number of missing entries was $l$. We then selected all the genes from matrix $B$ that contained $l$ missing entries, and randomly sampled one gene from them, denoted $g_1$. We then found the positions of the $g_1$ vector that were missing values, and assigned N/A to the same positions of the $g_x$ vector. By carrying out this procedure for every gene in matrix $A$, we obtained a simulated matrix $A^*$ that contained $p$ percent missing values.

### 2.4.4 Parameter Setting

A preliminary simulation study was conducted to select the best parameter setting for each method. We tested different values of the tuning parameters at every missing percentage, by repeating the mask-impute procedure three times. The parameter values tested for the NL method were $k = 5, 10, 15, 20$; the values for the KNN method were $k = 5, 10, 15, 20$; the values for LLS were $k = 5$ percent, 10 percent, 20 percent, 30 percent, . . . . . . , 90 percent of the number of genes of the data set; the values for SVD were nPcs = 2, 4, 6, 8, 10, 15, 20, 30, 40 (up to the number of arrays in the data set). No parameter selection was done for the BPCA method, because it’s been established that the recommended value for the nPcs parameter, which is the number of samples minus 1, performs the best [7]. Other than the major tuning parameters, we relied on the default of each method for other parameters.

To measure the success of imputation and compare the performance between different parameter settings, we used the normalized root mean squared error (NRMSE),

$$nrmse(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \div \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^2}.$$  

It measures the deviation of the imputed values from the hidden true values. For each method, the parameter setting that achieved the lowest average NRMSE in the preliminary simulation study (Tables 1 and 2) was selected for the larger scale simulation study, in which we repeated the mask-impute procedure 50 times for every data set at each of the five missing percentage settings.
2.4.5 Comparing the Performance

To compare the performance of the methods, we employed two methods. The first was NRMSE. The second method measured the impact of the imputation on the statistical testing results. The general idea was to measure how much the imputation changed the list of significant genes in statistical testing, as compared to the test results generated from the original data without missing values. First, testing was conducted on the complete data, and the list of significant genes taken as the reference. Then after each mask-impute process, the same testing procedure was applied and the list of significant genes was compared with the reference list. The false positive (FP) and false negative (FN) counts were analyzed to judge the success of the imputation. It should be noted that the reference list from the original data was not the biological truth. Thus, the results measured only relative changes, and the FP and FN counts carried no direct biological interpretation. The idea was that the imputation method should change the list of significant genes as little as possible.

Different statistical tests were selected based on the characteristics of each data set. All the tests were done on a gene-by-gene basis. After testing, the p-values from all the genes were transformed to false discovery rate (FDR) [28], and genes with FDR less than a threshold (detailed in the Results section) were considered significant. Different FDR cutoffs were selected for different data sets, in order to reach a significant gene list of reasonable size for the assessment of FPs and FNs rates after imputation.

For the B-cell lymphoma profiling (Alizadeh) data set [21], the main clinical interest was the survival outcome. The accelerated failure time (AFT) model assuming the Weibull distribution was fitted for each gene using the samples with survival information. For the yeast transcriptome/translatome comparison (Halbeisen) data set [22], we fitted a multiple linear regression model for each gene, with RNA source (transcriptome/translatome) as one predictor, stress treatments as another predictor, and the gene expression value as the response. Because the main interest of the study was the contrast between transcriptome and translatome, we focused on genes significantly associated with the RNA source predictor. For the NCI60 cell line (Ross) data set [23], a regression model was fitted for each gene, with the tissue origin of the cancer cell lines as the predictor, and the gene expression value as the response. The F-test p-value for the overall fit, which measured the association of the gene expression with the tissue origin, was collected. For the yeast cell cycle time series (Spellman) data sets [25], because the main interest was in genes showing periodic behavior, the Fisher’s exact g test for periodicity [29] was used to identify periodically expressed genes. The only data set for which no test was performed was the GSE19119 (Tymchuk) data set [24], as no clear genes. The only data set for which no test was performed showing periodic behavior, the Fisher’s exact g test for periodicity, the Halbeisen data set [25], because the main interest was in genes associated with the gene expression with the tissue origin.

We illustrate the calculation of SCOL in Fig. 1. The data points are normal score transformed and ordered by the X values, and the absolute distance between adjacent Y values in the list are summed up (Fig. 1, red bars). After further transformation as described in the Methods section, the SCOL similarity measure is obtained. When the spread of Y|X is small (Fig. 1a), the distances are small and the similarity is high. When the spread of Y|X is large (Fig. 1b), the distances are big and the similarity is low. Clearly, SCOL similarity score is asymmetric as demonstrated by the two panels in Fig. 1 (same data; role of X and Y reversed). Because in missing value imputation, we only focus on the conditional variance of the gene with a missing value given the value of the informative gene, the asymmetric version of SCOL serves the purpose well. In the case of Fig. 1a, an observed value in X will be helpful to predict the value of Y. Yet in the case of Fig. 1b, an observed value in X will not be as helpful in predicting the value of Y.

3 Results and Discussions

We illustrate the calculation of SCOL in Fig. 1. The data points are normal score transformed and ordered by the X values, and the absolute distance between adjacent Y values
the FDR. Hence, an imputation method that doesn’t introduce too many false positives is preferable. In this aspect, NL clearly outperformed the other methods.

3.2 The Impact of Rowwise Mean Removal on the Evaluation of Imputation Methods

The most common method of evaluating the performance of imputation methods is through simulations using existing microarray data sets. However, it should be noted that some preprocessing of the data sets may cause artifacts in the evaluation. Particularly, overly optimistic results may be obtained for imputation methods that use global correlation structures, when the processing of the data introduces artificial dependency among the columns (arrays) of the data matrix. This is the case because although some data points are changed to “missing” in the simulation, their true values have played a role in the normalization already, and the information needed to retrieve the true values is in fact

![Fig. 2. Simulation results for four data sets based on 50 simulations. The data sets (labeled on the left side) are in rows and the type of results (labeled on top) are in columns. Error bars represent standard errors. No testing was performed for the Tymchuk data, because no clear outcome can be used for testing.](image-url)
buried in other columns. In other words, with the row-wise normalization, each column of the matrix becomes perfectly linearly dependent on other columns. If an imputation method utilizes this dependency, it will recover the missing value almost perfectly when there’s only one value missing in a row. However, this perfect recovery only happens in simulations. When a value is truly missing, it cannot have played a role in the normalization. With random noise in microarray measurements, perfect linear dependency between columns doesn’t exist in the raw data.

The Spellman’s cell cycle data were widely used in the evaluation and comparison of imputation methods. It has been the data set that showed the most dramatic differences between methods. When downloaded in matrix form at the cell cycle study website (http://cellcycle-www.stanford.edu/), the data were already subjected to normalization including row-wise mean removal within every time series experiment, i.e., every gene’s expression levels were normalized to obtain mean zero within every time series. This is suggested in the original publication [25] and evident by examining the data. The data were rounded to two decimal places, hence the row means can deviate very slightly from zero. When we used the two normalized Spellman time series data sets in the simulations, LLS and BPCA were dramatically better in terms of NRMSE than the other three methods (Fig. 3, column 1), which was consistent with previous studies [1], [5], [7].

In order to make a comparison, we downloaded the raw data (original log ratios) from the Stanford Microarray Database [30], and assembled the raw data matrix. When a gene was represented by more than one probes on the array, the mean of log ratios was taken to obtain a single reading. We then took the genes that were included in the simulation study with normalized data. Due to small differences in missing patterns, the numbers of genes were slightly different from the normalized data (Table 2). Using the non-normalized log ratios, we conducted another simulation study (parameters listed in Table 2). The results showed that with non-normalized data, although LLS and BPCA still led in performance in NRMSE, the difference between the methods was drastically smaller (Fig. 3, compare column 2 with column 1). To prove the point from the other end, we conducted another simulation using the NCI60 data. To introduce a dependency that is at similar level as the cell cycle data, we partitioned the NCI60 data matrix into three equal-sized submatrices, by selecting three groups of continuous columns—columns 1-20, 21-40, and 41-60. Within every submatrix, we conducted row-wise mean removal. Then, we subjected the whole matrix to the mask-impute simulation procedure. The simulation yielded a dramatic decreases in the NRMSE from LLS and BPCA (Fig 3, row 3). Comparatively, the SVD method, although utilizing global dependency, did not make as much artificial gains when the data were row-normalized. This is because although the last three eigen values were driven to zero by row-normalization, the relative scale of the larger eigen values didn’t change much. The SVD method largely relies on the first several singular vectors, which prevented it
from taking advantage of the perfect dependency created by row-wise mean removal. Based on these results, we can see that the impact of row-wise mean removal on the NRMSE results could be substantial. In addition to influencing the performance estimations in simulation studies, normalization can also impact real-data analysis, in which we often resort to cross validation using the complete portion of the data to select the method and parameter settings. The same artifact as detailed above could influence the choice of method and parameter settings, if the datum is subjected to row-wise mean removal before the cross validation.

Although LLS and BPCA didn’t show as much advantage using the original data, they still led in the performance as judged by NRMSE. This is expected because the cell cycle arrays contain strong between-array dependencies [1]. We further explored whether the better NRMSE results translated to better testing results. Here, we used the Fisher’s exact g test to find periodic behaviors in the genes, because the Spellman experiment was designed to find cell cycle related genes. The FDR cutoff of 0.2 was used, which yielded ~10% significant genes in both data sets (Supporting Table 1). NL and KN showed a clear lead in FP + FN counts despite they trailed in NRMSE results. Again NL was the most conservative method in terms of FP/FN ratio, though for the elutriation data set all method yielded many more false positives than false negatives (Fig. 3, column 4).

To better understand why NL tended to be more conservative in terms of FP/FN ratio, we selected one of the scenarios, 10 percent missing in the Halbeisen data set, for further examination. We repeated the mask-impute process 16 times and recorded the test p-values for all the genes. We then compared the test p-values after imputation against the complete data p-values. For the NL method, 46.2 percent of the test p-values after imputation were smaller than the corresponding p-values from the complete data, while for all the other methods, over 50 percent of the test p-values after imputation were smaller than the corresponding p-values from the complete data (KNN: 52.6 percent, BPCA: 54.4 percent; LLS: 53.2 percent; SVD: 55.8 percent). Given that most of the genes were insignificant in the complete data, the tendency of the methods to decrease the p-values, even slightly, made a lot of false positives. NL was conservative in the sense that on average it tends to decrease the significance of a gene, albeit slightly, when imputation is done.

In summary, the nonlinear imputation method based on SCOL and kernel smoother showed strong performance in the imputation of missing data, especially when judged by the impact on the statistical testing results. Aside from generating lower FP + FN counts, its property of being conservative in terms of FP/FN ratio is desirable, as controlling false discovery rate is a key concern when conducting tests using high-throughput data.

ACKNOWLEDGMENTS

The authors thank Dr. Qi Long for helpful discussions. This research is partially supported by NIH grants 1P01ES016731-01, 2U19AI057266-06, 5P30AI50409-10, and 1UL1RR025008-02, and a grant from the University Research Committee of Emory University.

REFERENCES


Tianwei Yu received the PhD degree in statistics from the University of California, Los Angeles, in 2005. Since 2006, he has been an assistant professor in the Department of Biostatistics and Bioinformatics at Emory University. His current research interests include high-dimensional data analysis, biological networks, and data preprocessing in metabolomics.

Hesen Peng received the BS degree in statistics from Fudan University, Shanghai, China, in 2008, and is currently a second year PhD degree student in biostatistics at Emory University. In 2008, he was a research intern at Chinese Academy of Science—Max Planck Group Partner Institute of Computational Biology. His research interests include high-dimensional data analysis, nonlinear multidimensional network discovery, and time series analysis.

Wei Sun received the PhD degree in statistics from the University of California, Los Angeles, in 2007. Since 2007, he is an assistant professor in the Department of Biostatistics and the Department of Genetics, University of North Carolina at Chapel Hill. His current research interests include developing statistical/computational methods and software in the area of statistical genetics or genomics. In particular, he is interested in the topics of gene expression QTL (eQTL), RNA-seq and allele-specific expression, ChIP-chip/ChIP-seq analysis, and copy number alterations.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.