LINEAR STEPWISE REGRESSION (LSR) IMPUTATION AS A NOVEL METHOD FOR ESTIMATION OF THE MISSING VALUES IN GENE EXPRESSION PROFILE

Mou'ath A. HOURANI[1], Ibrahiem M. M. EL EMARY[2]
[1] Faculty of Information Technology, Al Ahliyya Amman University, Amman 19328 Jordan
[2] Faculty of Engineering, Al Ahliyya Amman University, Amman 19328 Jordan
e-mail: [1]mouath.hourani@gmail.com, [2]omary57@hotmail.com

Keywords: Linear Stepwise Regression (LSR), Analysis of variance (ANOVA), *p*-value, cDNA microarray data, gene expression profile.

Abstract: *In this paper, an innovative missing value estimation algorithm called Linear Stepwise Regression (LSR) is presented which uses multiple correlated-based samples imputation matrices for the final prediction of missing values. The matrices are computed and optimized using linear stepwise regression and linear programming methods. The performance of the LSR impute method, assessed over five different data sets, has been compared with four imputing approaches, namely KNN, LSS, LSimpute3 and LSimpute5 impute methods. Testing results reveal that the LSR impute has outstanding prediction ability in the estimation of the missing values problem for some data sets and is robust against the increasing rate of missing values. A comprehensive comparison of NRMSE on five data sets shows that the LSR impute performs comparative with, if not better than, the other missing value estimation methods in this area, and when complemented with other leading methods, it appears to be a proper solution to the missing value estimation in gene expression profile. Finally, our LSR method is applicable over other various non-bioinformatics data.*

INTRODUCTION

In spite of containing a considerable numbers of missing values, microarray data are used in a range of application areas in biology. These missing values can significantly affect subsequent statistical analysis and machine learning algorithms, so there is a strong motivation to estimate these values as accurately as possible. While many imputation algorithms have been proposed [1][2][3][4], more robust techniques need to be developed so that further analysis of biological data can be accurately undertaken. In this paper, an innovative missing value estimation algorithm called Linear Stepwise Regression (LSR) is presented which uses multiple correlated-based samples imputation matrices for the final prediction of missing values. The matrices are computed and optimized using linear stepwise regression and linear programming methods. The LSR algorithm builds a model based on grouping the strongly correlated samples into clusters and it imputes the missing values using linear stepwise regression algorithm. LSR is a complementary algorithm which can be used in parallel with any leading imputation method. Unlike other methods, LSR transforms the genes/samples from the original domain into a new domain called R-domain[1] which in turn does not have the limitations of the distance matrix-based algorithms. Furthermore, LSR implements the idea of linear stepwise regression to assign the weight into the missing values.

Five different data sets are used in this paper for the application of microarray. The reason behind using these data sets is because

[1] see http://CRAN.R-project.org for a full R language discussion.

of their type varieties. These sets contain data with different characteristics, such as time series [5], noisy (Schizophrenia data set) [6], highly correlated data (PD data set) [7], cancer disease data (CCDATA) [5] and absence of patterns (Niehs data set) [8] [9]. By using these data sets under different imputation methods, we are trying to assess the performance of the LSR algorithm compared with other methods. In this sense, the use of any imputation method for a specific data type should be carefully determined. Given the range of data set types and the limitations in current algorithms to handle different types of data sets, we introduce the LSR algorithm.

To perform missing value imputation using LSR, three steps are followed. First, the experiments are grouped (clustered) in such way that highly correlated experiments are clustered within the same group. The grouping is carried out by an ANOVA test [10] [11]. Second, instead of using average or zero values to create a complete matrix, we use LSimpute, LLS or any other algorithm. In this paper, we obtain two LSR3 and LSR5 algorithms when applied over LSimpute type 3 and 5 (we call them LSimpute3 and LSimpute5 respectively, see Bo et al (2004) [12] for more details). Finally, multilinear regression (specifically, linear stepwise regression) is used to build the model and impute the missing values.

A comparative study of our method with the previously developed methods, including the KNNimpute [1], LLSimpute [3] and LSimpute [12] methods has been presented for the estimation of the missing values on five gene expression data sets. Among different algorithms we compared, the LSR5 algorithm obtained better or at least comparable estimation results with small Normalized Root Mean Square Error (NRMSE) on different kinds of data sets. The outstanding estimation ability of this imputation method is partly due to the efficient use of the missing value information feature that exists in a multilinear regression scheme. This paper consists of nine main sections. In the first three sections, we illustrate in details the Linear Stepwise Regression (LSR) method. In Section 5, we introduce our proposed model. Experimental results are presented in sections 6 and 7. Finally, in Sections 8 and 9 the discussions and conclusions are presented and outlined.

1. LINEAR STEPWISE REGRESSION (LSR) IMPUTATION METHOD

The LSR method is a data mining and statistical technique that uses multilinear regression to model a linear equation for imputation purposes [10][14][15]. It is a data mining method, in the sense that it groups the most correlated samples into clusters. Furthermore, it is a statistical method in the sense that it uses linear regression to build the model. LSR consists of three main steps: Analysis of Variance (ANOVA) test, which is used to group the most correlated samples; Stepwise regression to build the model; and stop criteria to optimize the solution. In the subsequent sections we explain each step in detail.

2. ANALYSIS OF VARIANCE

The Analysis of Variance (ANOVA) is a tool that tests the difference between the means of two or more groups [10] [11]. A one-way ANOVA or single factor ANOVA tests differences between groups that are only classified on one independent variable.

The advantage of using ANOVA rather than multiple t-tests or any other statistical tool is that it reduces the probability of a type-I error (the error of rejecting a null hypothesis when it is actually true - this is the error of accepting an alternative hypothesis) [11][13]. Making multiple comparisons increases the likelihood of finding something by chance-making a type-I error.

In microarray analysis literature, ANOVA test is used as a feature selection and classification tool [8]. There are two possible ways to deal with microarray data in terms of classification. All genes can be used to classify a small number of samples in distinct classes or all samples can be used to classify a large

number of genes in distinct classes. For example, in a two-dimensional space it is always possible to separate two observations perfectly; the same thing is true for three or fewer points in a three-dimensional space and for n or fewer points in an n-dimensional space. This observation has been explained in Amaratunga and Cabrera (2004) [8].

ANOVA, on another hand, is also used as a feature selection tool. By finding the genes with the small *p*-value that are differentiated between two groups of samples, we can state that the selected *p*-values are the most differentiated ones in this experiment. For the case of only two classes, this approach is equivalent to finding the genes with the small *p*-value in an ordinary two-sample t-test [17]. Dudoit and Fridlyand (2003) [18] restrict their attention to a mixed number of genes that have the highest ratio of between-to within-groups sums of squares. This corresponds to taking the genes with the smallest *p*-value in an ordinary one-way ANOVA setting. Nguyen and Rocke (2002) [17] and Radmacher et al. (2002) [19], describe similar gene filtering approaches. Most authors select a pragmatic number of genes so as to make the number of variables smaller than the number of samples, whereas Amaratunga and Cabrera (2004) [16] fix a certain significance level and reduce the number of genes by selecting a number of principal components, which is again smaller than the number of samples. In this paper, the Dudoit and Fridlyand (2003) [18] approach is used.

The main purpose of applying a column-based method is that it results in a traditional classification problem, where the number of samples is less than the number of genes. Furthermore, there are a number of problems associated with gene-based approach [16] [17]. First of all, there is the issue of bias. For example, in a completely random gene expression data set with thousands of genes and a relatively small number of samples, this gene selection procedure will find \statistically significant" genes. These genes, thus selected, are expected to perform well in classifying the data set. Nevertheless, the classification does not have any predictive power. The only way in which this procedure is valid is if the genes are selected only on the basis of the training set rather than on the full data set and if a validation set is used to estimate the misclassification rate. If the significant genes were selected on the basis of the whole data set and subsequently reused for estimating the misclassification rate, a far too noisy picture of the procedure would be painted.

The LSR uses an ANOVA test to find a subset (group) of the most differentiated samples for each specified missing value at a time. As mentioned before, the ANOVA test returns the most differentiated samples for each missing value, and these samples represent the highest *p*-values.

3. THE MULTIPLE REGRESSION MODEL

Multiple regression analysis is a general statistical technique used to analyze the relationship between a single dependant variable and several independent ones [13] [11]. In the multiple model it is assumed that a linear relationship exists between the dependent variable Y, and n independent variables, X1, X2,..., Xn. The multiple regression model equation can be represented by:

$$Y = X_1 + X_2 + \ldots\ldots + X_n \qquad (1)$$

The objective of multiple regression analysis is to use the independent variables, whose values are known, to predict the single dependent value selected [11]. Each independent variable is weighted by the regression analysis procedure to ensure maximal prediction from the set of independent variables to the overall prediction.

In multiple regression, the linear regression model for determining y given x can be represented as $y = \alpha + \beta x + e$, where e is the error term for which the variance is minimized when estimating the model with least squares [20]. In single regression, α and β (model parameters) are estimated using $\hat{\alpha} = \hat{y} - \hat{\beta}\bar{x}$ [20] and

$$\hat{\beta} = \frac{S_{xy}}{S_{xx}}, \qquad (2)$$

where $S_{xy} = \frac{1}{n-1}\sum_{j=1}^{n}(x_j - \bar{x})(y_i - \bar{y})$

is the covariance between *x* and *y*,

$$S_{xx} = \frac{1}{n-1}\sum_{j=1}^{n}(x_j - \bar{x})^2 \qquad (3)$$

is the variance of *x*, and *n* is the number of observations. Here $\bar{x}$ and $\bar{y}$ are the averages over $x_1,\ldots\ldots, x_n$ and $y_1,\ldots\ldots, y_n$. Therefore, given a variable *x*, the least squares estimate of a variable *y* can be written as [20]:

$$\hat{y} = \bar{y} + \frac{S_{xy}}{S_{yy}}(x - \bar{x}). \qquad (4)$$

To create a multiple regression model for $y_1,\ldots\ldots, y_l$ given $x_1,\ldots\ldots, x_k$, we have $y_i = \alpha_i + \beta_{i1}x_1 + \beta_{i2}x_2 + \ldots\ldots + \beta_{ik}x_k + e; i = 1,\ldots\ldots l.$ It can be shown that the least squares estimate for this model can be formulated as [20]

$$\hat{y} = \bar{y}_i + S_{y_i} x S^{-1} x x (x - \bar{x}) \qquad (5)$$

where:

$\boldsymbol{X} = [x_1, x_2, \ldots\ldots, x_k]^T, \bar{\boldsymbol{X}} = [\bar{x}_1, \bar{x}_2, \ldots\ldots, \bar{x}_k]^T,$
$S_{y_i} x = [S_{y_i x_1}, S_{y_i x_2}, \ldots\ldots, S_{y_i x_k}]$ and

$$\mathbf{S}_{xx} = \begin{pmatrix} S_{x_1x_1} & \ldots\ldots & S_{x_1x_k} \\ \ldots\ldots & \ldots\ldots & \ldots\ldots \\ S_{x_kx_1} & \ldots\ldots & S_{x_kx_k} \end{pmatrix}$$

The single regression model has two parameters to be estimated, while the multiple regression model has $l(k + 1)$ parameters [20].

4. BUILDING THE MODEL

4.1 BUILDING THE MODEL: CRITERION FOR MODEL SELECTION

The ANOVA test produces a set of predictors - number of columns to be used to estimate a specific missing value at a time. From any set of *p* predictors chosen, 2^p alternative models can be constructed [20][15][14]. This calculation is based on the fact that each predictor can be either included or excluded from the model. To build a model, there is the regression model with no *X* variables, i.e., the model $\boldsymbol{Y_i} = \beta_0 + \varepsilon_i$. Then there are the regression models with one *X* variable (X_1,X_2,X_3,X_4), with two *X* variables (X_1 and X_2; X_1 and X_3; X_1 and X_4; X_2 and X_3; X_2 and X_4; X_3 and X_4), and so on.

In most circumstances, it will be impossible to make a complete examination of all possible regression models. For instance, in our experiment there are 30 potential *X* variables in the pool and thus 2^{30} possible regression models. Generally, this is a very time-consuming process.

Model selection procedures or variable selection procedures have been developed to identify a small group of regression models that are good according to a specified criterion [20] [15] [14]. A detailed examination of a limited number of promising models will lead to the selection of the final regression model to be employed.

Many criteria for comparing the regression models have been developed. As we mentioned, if there are *p* potential predictors, then there are 2^p possible models. We fit all these models and choose the best one according to some criterion. The Akaike Information Criterion (AIC) and the Schwarz' Bayesian Information Criterion (BIC) are the most commonly used criteria. We search for models that have small values of AIC or BIC, where these criteria are given by [20] [10] [14]:

$$AIC = n \ln SSE - n \ln n + 2p \qquad (6)$$

while

$$BIC = n \ln SSE - n \ln n + [\ln n]p \quad (7)$$

Where *m* is the number of entire columns and SSE denotes to error sum of squares. For linear regression models, we want to minimize AIC or BIC. Larger models will fit better and so have smaller error sum of squares (SSE) but use more parameters. Thus the best choice of model will balance fit with model size [10]. BIC penalizes larger models more heavily and so will tend to prefer smaller models in comparison to AIC [10]. AIC and BIC can be used as selection criteria for other types model too.

4.2 BUILDING THE MODEL: SEARCH PROCEDURES FOR MODEL SELECTION

As noted in the previous section, the number of possible models grows rapidly with the number of predictors. Evaluating all of the possible alternatives can be a time-consuming process. To simplify the task, we will use an automatic search procedure called *stepwise regression* for selecting the model.

Stepwise Regression Procedures

Stepwise regression is the most widely used strategy for selecting independent variables for a multiple regression model. The procedure consists of a series of steps. At each step of the procedure each variable in the model is evaluated to see if, according to AIC and BIC criteria, it should remain in the model.

To build a regression model based on n independent observations of a response variable *Y* and a large set of p potentially useful predictors $X_1, X_2, \ldots, X_p$, a sequence of approximating models $,y_1, y_2, \ldots, y_p$, should fit into the form [20][10][14]:

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots\ldots + \beta_k x_k + e \quad (8)$$

For each choice of k, the chosen model ideally minimizes the sum of squared residuals,

$$SSE_k = \sum_i (Y_i - \hat{Y}_i, K)^2 \qquad (9)$$

among all models with *k* predictors.

Suppose, for example, that we wish to perform stepwise regression for a model containing the *p* predictors variables (samples) obtained from the ANOVA process. The criterion measure (see equations 6 and 7) is computed for a model containing *p* predictors variables/samples. The criterion measure is computed for each sample, and of all the samples that do not satisfy the criterion are removed from the model. If a sample is removed in this step, the regression equation for the smaller model is calculated and the criterion measure is computed for each sample in the model. If any of these samples fail to satisfy the criterion for inclusion in the model, the one that least satisfies the criterion is removed. If a sample is removed at this step, the sample that was removed in the first step is reentered into the model, and the evaluation procedure is continued. This process continues until no more samples can be inserted or removed. The main advantage of the stepwise regression is that if a sample is deleted from the model in one step, it could be evaluated for possible reentry into the model in subsequent steps [20][10][14]. Our LSR proposed algorithm that explains all previous steps is presented in the next section.

4.3 BUILDING THE MODEL: THE ALGORITHM

Impute Algorithm (M,Max,Ref,Output)

Input:

- *M* : Matrix (*m* rows and *n* columns) of real with missing Values
- *Max* : Maximum number of columns allowed in the regression models
- *Ref* : Matrix (*m* rows and *n* columns) with an initial estimation for the missing values

Output:

- *Out* : Matrix with Missing values imputed

Begin

- **For** each column $M^{(i)}$ in *M*
  - **For** each column $Ref^{(j)}$, $j \neq i$
    - ⋆ Calculate the absolute value of Pearson's correlation C_{ij}
  - **end For**
  - *S* ← Select Max Columns from *Ref* with the highest values for C_{ij}
  - If number of columns in *Ref* is less than *Max* Then use all columns except column *i*
  - **end If**
  - *LM* ← Initial Linear Model $M^{(i)}$ as function of the columns in *S*
  - *LM* ← *StepWiseP rocedure*(*LM*)
  - **For** each missing value in $M^{(i)}$
    - ⋆ estimate the missing value M_{ik} using the linear Model LM
    - ⋆ store the imputed values in *Out*
  - **end For**
- **end For**
- Write the output matrix *Out*

End

5. EXPERIMENTAL RESULTS

5.1 DATA SETS

The performance of each method for predicting missing values is evaluated by using five cDNA microarray experiments data sets.

The first data set is called Niehs. It is based on a study of human cell lines. This data set is composed of three dye-swaps, thus six arrays. The data are from the Niehs experiments comparing treated and control human cell lines, as described in Kerr *et al*. (2002) [8] [9]. It is publicly available at http://www.jax.org/staff/churchill /datasets/expression/niehs. In the Niehs data set there are 1,907 genes and no missing values, thus a full intensity data matrix of dimension 1,907 × 12.

The second example is gene expression data from the study of Schizophrenia disease. This data set is from Bowden *et al.* (2005) [6], and has been generated in Newcastle University, Australia. It is composed of 14 non-psychiatric control individuals and 14 patients diagnosed with schizophrenia, matched for age and gender. All participants in the study have no recent history of substance abuse, as there is controversy about the effects that certain drugs have in schizophrenia. The original data file contained 6,000 genes, after removing genes with one or more missing values, the resulting gene expression profile contained 2,901 genes × 14 experiments. More details are available from Bowden *et al.* (2005) [6].

The third data set example is gene expression data from typical studies on primary tumors (CCDATA) [5]. The CCDATA data set is based on samples from cervical tumors before and after radiotherapy and is composed of 16 dye-swaps and thus 32 experiments arrays. In the original cervical cancer data set, 22% of the data were missing, affecting 70% of the 14,229 genes. We have removed the genes with one or more missing values, leaving 4,246 genes. The resulting intensity data matrix has 4,246 genes × 64 experiments. The data is available http://genome-www.stanford.edu/listeria/gut/.

The fourth data set is from an infection time series study [5]. Here we downloaded all the time course data and removed all genes with missing values, resulting in a 6,850 × 39 data matrix. The data are available http://genomebiology.com/2002/4/1/R2.

The last data set is gene expression data from a study of Parkinson disease (PD) introduced in Brown *et al*. (2002) [7]. In the original file, 17% of the data were missing, affecting 30% of the 9,000 genes. We have removed the genes with one or more missing values, leaving data from 5,636 genes. The resulting intensity data matrix is of dimension 5,636 × 80. More detailed information about this data can be found in Brown *et al*. (2002) [7].

The data sets we used in our study went through several processing steps. Firstly, they were log-transformed after being taken from the image (i.e. after normalization). Secondly, the rows and the columns which contained too many missing values (i.e., 10% and more) were excluded. Thirdly, before using the LSR method, each of the columns was scaled to between 0 and 1, which means the data sets are normalized. Mean-normalizing the data will further help in regression performance using Euclidean Distance. Finally, the data sets with these pre-processing steps were used to construct the complete matrix.

Measurements of performance

In order to evaluate the performance of the missing value estimation methods, we constructed the complete matrices by removing all the rows containing missing values, and randomly created the artificial missing values, from 10% to 25% of the entries in a matrix. The artificial missing entries were introduced in two different ways:

Row-based: Randomly select a specific percentage of the entries in the complete matrix, and remove them. Between 10%-25% are removed in each row.

Column-based: Randomly select a specific percentage of the entries in the complete matrix, and remove them. Between 10%-25% are removed in each experiment/sample. Column-based method results are only shown in this paper.

The performance of the missing value estimation is evaluated by normalized root mean square error (NRMSE).

6. RESULTS

Table 1 shows the comparison of performance between the imputation methods. The results of applying four different methods on five data sets are shown. In this Table, the results reveal that LSR5 method always outperforms the LSR3 method. For example, when the percentage of entries missing is 20%, the NRMSE of the LSR5 reaches 0.10395, and the NRMSE of the LSR3 method is 0.12418 for Niehs data. Figures 1 to 5 show the performance of the six different methods on the five different data sets. The horizontal and vertical axes indicate the percentage of entries missing in the complete matrix and the NRMSE of each input scheme, respectively.

Performance comparison with other methods

Instance		Techniques					
		LSR 3	LSR 5	LS 3	LS 5	LLS	KNN
1907x12	Niehs 10%	0.10741	0.093610	0.15942	**0.09252**	0.10234	
1907x12	Niehs 15%	0.12418	**0.10395**	0.17812	0.10508	0.12112	
1907x12	Niehs 20%	0.15481	0.13911	0.20714	0.13920	**0.13408**	
2901x12	Schi 10%	0.98052	0.98389	1.03791	0.97455	**0.95456**	0.96575
2901x12	Schi 15%	0.90436	0.89960	1.00851	0.90453	**0.88912**	0.91175
2901x12	Schi 20%	0.92459	**0.92031**	1.09819	0.92283	0.975262	
4246x64	CCData 10%	0.18637	0.17479	0.26849	**0.17312**	0.32889	0.80094
4246x64	CCData 15%	0.19547	**0.18220**	0.27397	0.183401	0.34484	0.79821
4246x64	CCData 20%	0.20293	**0.18748**	0.27530	0.18868	0.36769	
4246x64	CCData 25%	0.21076	**0.194508**	0.28147	0.19788	0.38967	
6850x39	TS 10%	0.26452	0.26345	0.29476	**0.25961**	0.34111	0.49093
6850x39	TS 15%	0.26705	0.26569	0.29483	**0.26098**	0.34975	0.73361
6850x39	TS 20%	0.26465	0.26266	0.29469	**0.25816**	0.35355	
6850x39	TS 25%	0.27736	0.27552	0.31029	**0.27102**	0.37767	
5636x80	PD 10%	0.68565	0.68559	0.70847	0.67132	0.78062	**0.45583**
5636x80	PD 15%	0.68027	**0.66629**	0.70515	0.67958	0.77299	0.84692
5636x80	PD 20%	0.68453	**0.67165**	0.71104	0.68359	0.78844	
5636x80	PD 25%	0.68607	**0.67312**	0.71182	0.68563	0.79359	

Table 1 Comparison of basic LSR3 and LSR5 methods against KNNimpute, LSimpute3, 5 and LLSimpute with 10% - 25%.

The performance of the LSR impute method, assessed over five different data sets, has been compared with four imputing approaches, namely KNN, LLS, LSimpute3 and LSimpute5 impute methods. The K-value in the KNN impute method was preset as 15, according to the recommended range of between 10 and 20 [1], and both LSimpute3, 5 and the LLS impute methods are non-parametric methods so they do not require K-value. Performance of each method on different data sets is shown in Figures 1 to 5.

Niehs data is a challenging prediction data set, where a clear expression pattern is often absent [8]. Figure 1 shows among all other methods, the LSR5 method gets comparable NRMSE values. When the percentage of missing values in the data set is 15%, the LSR achieves the best result. And when the percentage of the missing values reaches 20%, the NRMSE of the LSR is a little larger than LLSimpute method but still smaller than that of the KNNimpute method and LSimpute3. This shows the LSR method is comparable with, if not better than, the previous methods on this data set.

The Schizophrenia and time series data (TS) were pre-processed by removing all genes containing the missing values. Because our experiments are based on sample imputation, no samples were removed in this experiment, even the ones that contain considerable missing values rate. From Figure 2 and 3 we can see that the LSR5 impute method notably starts to outperform the other methods when the missing rate is increased especially on the Schizophrenia data set. However, when we apply LSR5 on TS data, the NRMSE of LSR5 is a little larger compared to LSimpute5. Generally, the LSR performs stably across the noisy data.

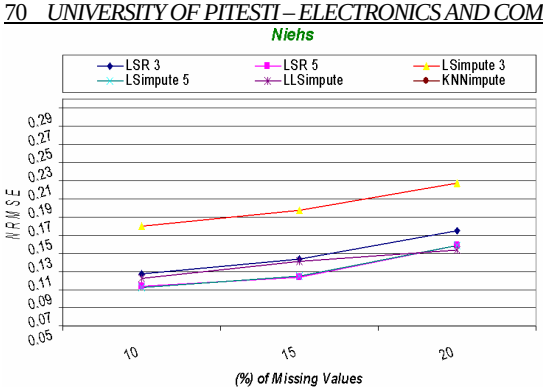

Fig. 1 Performance of the six methods on Niehs data. The percentage of entries missing in the complete matrix and the NRMSE of each missing value estimation method are shown in the horizontal and vertical axes, respectively.

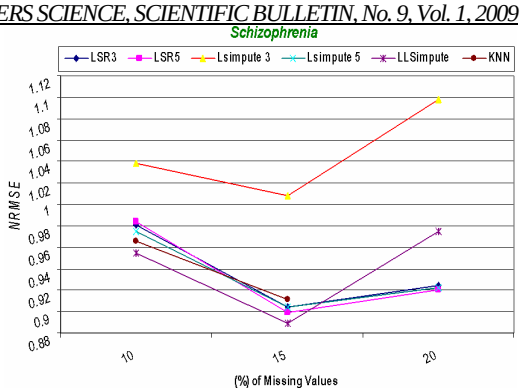

Fig. 2 Performance of the six methods on Schizophrenia data.

Relevant to many kinds of human cancers, including colorectal, ovarian, breast, prostate, as well as leukemia and melanomas, which involve much more complex regulation mechanisms, CCDATA human cancer data requires more reliable algorithms for missing value estimation. Figure 4 shows the performance of each method on this data set. In this case, the LSR5 method outperforms the other methods, especially when the missing rate is increased. For example, all the other methods get the estimate performance with the NRMSE between 0.19788 and 0.38967 for 25% missing, whereas our method is 0.19451. Consequently, the LSR5 impute method performs robustly as the percentage of the missing values increase.

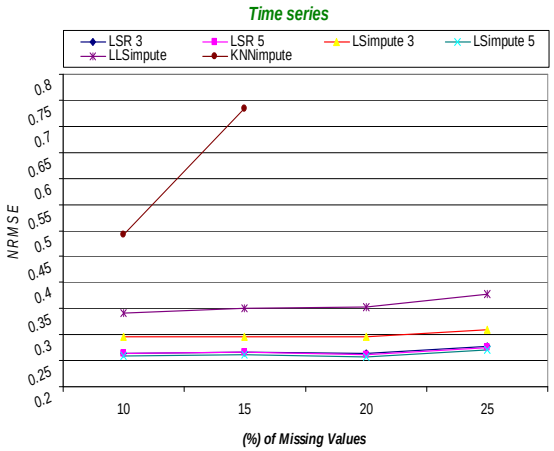

Fig. 3 Performance of the six methods on TS data.

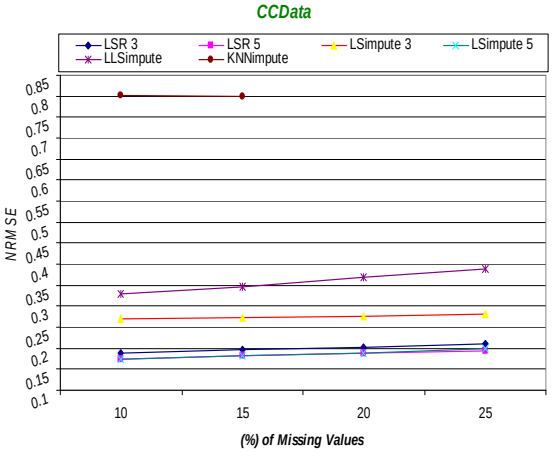

Fig. 4 Performance of the six methods on CCDATA data.

The PD data is used to test how much an imputing method is able to take advantage of strongly correlated genes in estimating the missing values [7]. We can see from Table 1 and Figure 5 that the LSR5 method outperforms other previous methods. However, in terms of memory and running time performances, the LSR5 method can take better use of strongly correlated genes than do the other four methods in estimating the missing values.

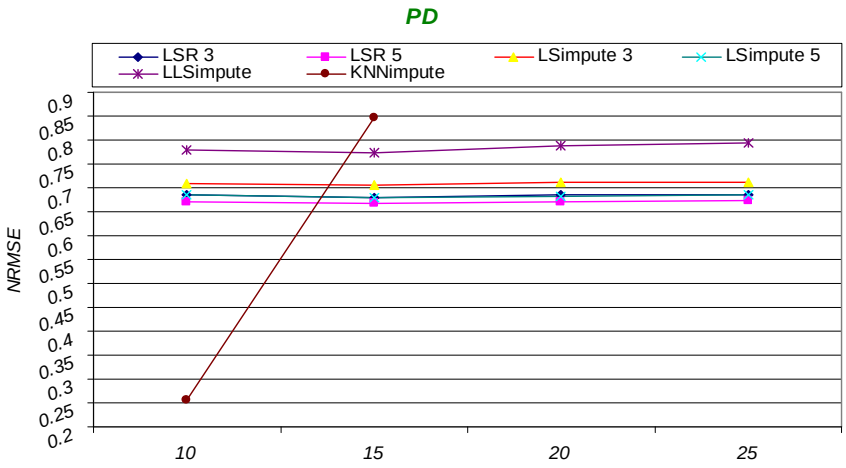

Fig. 5 Performance of the six methods on PD data.

7. DISCUSSION

Four existing imputation methods are used to evaluate the performance of the LSR impute method in our research. One of the advantages of the LSR method is that it makes most use of the information from the original data sets. The stepwise regression raises the estimation performance notably, which contributes to the best performance of the LSR method among these methods. In the case of the KNN and the LLS method, the redundant missing values in the samples with many missing values are just neglected, while the LSimpute simply regards them equally when modeling the missing values. Another advantage comes from the LSR method itself. The LSR method is a method based on the structural minimization principle (SMP is a family of statistical models that seek to explain the relationship among the variables. In doing so, it examines the structure of interrelationships among multiple variables) in which the global optimal solution is guaranteed [13][11]. The KNN method linearly combines the similar genes by weighting the average values of them. The coefficients used in combinations are calculated by using Euclidean Distance, which is not an optimal measurement for gene or sample similarity. This makes the KNN method perform worst among all the methods. The LLS and LSimpute are methods based on linear similarity structure. They share the similar linear combination of k-nearest genes as the KNN impute, and surpasses the KNN impute by optimizing the coefficients of the non-missing part of the similar genes using the least square solution. The LLS and LSimpute methods are based on local similarity structure of the data set, which draws back its performance when the local similarity is not very clear. In most cases, the LLS method performs worse than LSimpute5 but better than LSimpute3.

Besides the PD highly correlated data, our method also works well on the data sets those are more difficult for regression-based methods, because of the complex regulation mechanisms involved as in the case of CCDATA (Figure 4). Furthermore, the length of the expression profiles in PD data is 80 experiments, which is larger than the experiments in the other data sets (LSR is not affected by the increase in the number of sample/experiments as does by most other methods). This will make it more complex for regression. On the other hand, Figure 3 shows that the LSR5 method achieves comparative results to the previous methods. When the percentage of missing values becomes too large, the LSR impute

method performs little worse than do the LSimpute5. This is partly due to the stepwise regression search strategy for the parameters sets (the number of samples that are chosen form ANOVA step, see previous sections for more details). To maintain proper parameters sets (number of samples), the user should specify the range of the parameters being searched, so the parameters set might not be the optimum. The parameter selection is also a problem that has to be solved in the linear stepwise regression. Even if the parameter set might not be optimum, the result is still comparative with other impute methods. Thus the LSR impute method performs well in present research.

Finally, using any imputing algorithm requires the creation of a complete matrix. Calculating a complete matrix can be carried out by using average, zeros or ones as in the case of KNN, LLS and LSimpute. However, this will cause degradation in the performance of the final algorithm results. LSR algorithm uses a leading algorithm (LSimpute is used in this paper) to create the complete matrix which in turn increases the chances of getting more reliable results. However, if the number of samples in microarray is small, the performance of LSR declines. Consequently, we do not recommend using LSR method over 25% missing and if the number of experiments is less than 15.

CONCLUSIONS

In this paper, we introduced the Linear stepwise regression (LSR) imputation as a novel method for estimation of the missing values in gene expression profile. Testing results reveal that the LSR impute has outstanding prediction ability in the estimation of the missing values problem for some data sets and is robust against the increasing rate of missing values. Moreover, our approach makes most use of the missing value information in the whole gene expression matrix by restricting the attention to a fixed number of columns that have the highest ratio of between- to within-groups sums of squares (i.e. which corresponds to taking the samples with the smallest *p*-value in an ordinary one-way ANOVA setting).

A comprehensive comparison of NRMSE on five data sets shows that the LSR impute performs comparative with, if not better than, the other missing value estimation methods in this area, and when complemented with other leading methods, it appears to be a proper solution to the missing value estimation in gene expression profile. Finally, although our LSR method was examined using cDNA microarray data, applications to oligonucleotide array data, reverse transcription-polymerase chain reaction data, and others are obviously straightforward. Moreover, our method can be applied to various applications data.

REFERENCES

[1] TROYANSKAYA, O., CANTOR, M., SHERLOCK, G., BROWN, P., HASTIE, T., TIBSHIRANI, R., BOTSTEIN, D., ALTMAN, R.: Missing value estimation methods for DNA microarrays. Bioinformatics 17 (2001) 520-525

[2] NGUYEN, D., WANG, N., CARROLL, R.: Evaluation of missing value estimation for microarray data. Journal of Data Science 2 (2004) 347-370

[3] KIM, H., GOLUB, G., PARK, H.: Missing value estimation for DNA microarray gene expression data: local least squares imputation. Bioinformatics 21 (2005) 187-198

[4] WANG, X., LI, A., JIANG, Z., FENG, H.: Missing value estimation for DNA microarray gene expression data by support vector regression imputation and orthogonal coding scheme. BMC Bioinformatics 7 (2006)

[5] BALDWIN, D., VANCHIANATHAN, V., BROWN, P., THERIOT, J.: gene expression program reflecting the innate immune response of intestinal ephithelial cells to infection by listeria monocytogenes. Genome Biology 4 (2002) Available in http://genomebiology.com/2002/4/1/R2.

[6] WEIDENHOFER, J., BOWDEN, N., SCOTT, R., TOONEY, P.: Altered gene expression in the amygdala in schizophrenia: Up-regulation of genes

located in the cytomatrix active zone. Molecular and Cellular Neurosciences 31 (2006) 243-250

[7] BROWN, V., OSSADTCHI, A., KHAN, A., CHERRY, S., LEAHY, R., SMITH, D.: High-throughput imaging of brain gene expression. Genome Research 12 (2002) 244-254

[8] OUYANG, M., WELSH, W., GEORGOPOULOS, P.: Statistical analysis of a gene expression microarray experiment with replication. Statistica Sinica (12) 203-217

[9] KERR, M.K., CHURCHILL, G.A.: Experimental design for gene expression microarrays. Biostatistics 2 (2001) 183-201

[10] DANIEL, W.: Biostatistics A foundation for analysis in the health sciences. 8th edn. John Wiley and Sons (2004).

[11] KUNTER, M., NACHTSHEIM, C., NETER, J., LI, W.: Applied linear statistical models. 5th edn. McGraw-Hill Irwin (2005)

[12] BO, T., DYSVIK, B., JONASSEN, I.: LSimpute: accurate estimation of missing values in microarray data with least squares methods. Nucleic Acids Research 32 (2004)

[13] PAVKOV, W., PIERCE, K.: Ready, set, go! A Student guide to SPSS(R) 11.0 for Windows. 1st edn. McGraw-Hill (2003)

[14] SPICER, J.: Making sense of multivariate data analysis. 1st edn. SAGE Publication (2005)

[15] CRAMER, D.: Advanced quantitative data analysis. 1st edn. Open University Press (2003)

[16] AMARATUNGA, D., CABERA, J.: Analysis of data from viral DNA microchips. Journal of the American Statistical Association 96 (2001) 1161-1170

[17] NGUYEN, D., ROCKE, D.: Multi-class cancer classification via partial least squares with gene expression pro¯les. Bioinformatics 18 (2002) 1216-1226

[18] DUDOIT, S., FRIDLYAND, J., SPEED, T.: Comparison of discrimination methods for the classification of tumor using gene expression data. 576 (2000)

[19] HEDENFALK, I., Duggan, D., Radmacher, Y., Bittner, M., Meltzer, S., Gusterson, P., Esteller, B., Raffeld., M.: Gene expression profiles in hereditary breast cancer. The New England Journal of Medicine (344) 539-548

[20] Johnson, R., Wichern, D.: Applied multivariate statistical analysis. 5th edn edn. Prentice Hall (2002)