BIOINFORMATICS

Vol. 19 no. 17 2003, pages 2302–2307
DOI: 10.1093/bioinformatics/btg323

Missing-value estimation using linear and non-linear regression with Bayesian gene selection

Xiaobo Zhou[1], Xiaodong Wang[2] and Edward R. Dougherty[1,3,*]

[1] Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, USA, [2] Department of Electrical Engineering, Columbia University, New York, NY 10027, USA and [3] Department of Pathology, University of Texas M.D. Anderson Cancer Center, Houston, TX 77030, USA

Received on January 17, 2003; revised on February 23, 2003; accepted on June 11, 2003

ABSTRACT

Motivation: Data from microarray experiments are usually in the form of large matrices of expression levels of genes under different experimental conditions. Owing to various reasons, there are frequently missing values. Estimating these missing values is important because they affect downstream analysis, such as clustering, classification and network design. Several methods of missing-value estimation are in use. The problem has two parts: (1) selection of genes for estimation and (2) design of an estimation rule.

Results: We propose Bayesian variable selection to obtain genes to be used for estimation, and employ both linear and nonlinear regression for the estimation rule itself. Fast implementation issues for these methods are discussed, including the use of QR decomposition for parameter estimation. The proposed methods are tested on data sets arising from hereditary breast cancer and small round blue-cell tumors. The results compare very favorably with currently used methods based on the normalized root-mean-square error.

Availability: The appendix is available from http://gspsnap.tamu.edu/gspweb/zxb/missing_zxb/ (user: gspweb; passwd: gsplab).

Contact: edward@ee.tamu.edu

1 INTRODUCTION

Data from microarray experiments are usually in the form of large matrices of gene expression levels under different experimental conditions, and frequently there are missing values. The missing-value phenomenon occurs for various reasons, e.g. the *Drosophila* genes (Arbeitman *et al*., 2002), including insufficient resolution, image corruption or simply due to dust or scratches on the slide. Missing data may also occur systematically as a result of the robotic methods used to create them. Data may be missing on account of an image quality metric meant to delete low-quality spots (Chen *et al*., 2002).

One solution to the missing data problem is to repeat the experiment (Butte *et al*., 2001; Troyanskaya *et al*., 2001). This strategy can be expensive, but has been used to validate microarray analysis algorithms. Missing log-two or natural-log data may be replaced by zeros (Alizadeh *et al*., 2001) or by an average expression over the samples ('row average'). Two methods for missing-value estimation have been proposed by Troyanskaya *et al*. (2001): a singular value decomposition method (SVDimpute) and a weighted K-nearest neighbor method (KNNimpute). The KNNimpute method is proposed as a robust and sensitive method for missing-value estimation (Troyanskaya *et al*., 2001). It uses the KNN procedure to select genes, and uses weighted linear combinations to predict missing values. However, the genes selected by KNN are not necessarily among the best choices for linear prediction of the target gene because the gene selection and missing-value estimation are treated as two independent procedures, namely using two different models. With this as our motivation, we approach missing-value estimation from the viewpoint of linear or nonlinear regression with Bayesian variable selection (see Lee *et al*., 2003; Smith and Kohn, 1997, for Bayesian variable selection). Whereas variable selection is often used for class separability, here we need to find genes that are highly correlated with each other, which is akin to the cluster analysis problem.

In this study, we formulate the gene selection problem as a linear or nonlinear regression with Bayesian variable selection, and devise a Gibbs sampler to solve it. The proposed linear and nonlinear regression techniques with variable selection are computationally intensive. To mitigate the complexity, we develop some procedures for fast implementation of some key steps of the algorithms. We test our proposed methods on breast cancer data (Hedenfalk *et al*., 2001) and small round blue-cell tumor data (Khan *et al*., 2001). The results show that the linear and nonlinear regression with Bayesian gene selection offers substantially better estimation accuracy than the KNNimpute method in terms of the

[*] To whom correspondence should be addressed.

normalized root-mean-square (RMS) error for artificially introduced missing values. This is significant because, as shown by Troyanskaya *et al.* (2001), KNNimpute compares favorably with filling with zeros and row average, and performs similar to SVDimpute.

The paper is organized as follows. In Section 2, we develop a missing-gene prediction algorithm based on linear regression with Bayesian gene selection. In Section 3, we discuss some implementation issues including some fast algorithms for Bayesian gene selection. In Section 4, we develop a missing-gene prediction algorithm based on nonlinear regression with Bayesian gene selection. Section 5 provides experimental analysis and comparisons. Section 6 contains the conclusions.

2 LINEAR REGRESSION WITH BAYESIAN GENE SELECTION

2.1 Problem statement

Assume gene y has one missing value in the $(m+1)$-th experiment. Missing-value estimation should find other genes highly correlated with y, based on the results from experiments 1 to m, that have values present in the $(m+1)$-th experiment, and use them to predict the $(m+1)$-th value of y.

Assume there are $n+1$ genes, say $z_1, \dots, z_n, z_{n+1}$. Define a complete data set $\boldsymbol{Z} = (z_{ij})_{(m+1)\times(n+1)}$, i.e. $(m+1)$ experimental results for $(n+1)$ genes, which is denoted by

$$\boldsymbol{Z} = \begin{bmatrix} \text{Gene 1} & \text{Gene 2} & \cdots & \text{Gene } n & \text{Gene } n+1 \\ z_{1,1} & z_{1,2} & \cdots & z_{1,n} & z_{1,n+1} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,n} & z_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ z_{m,1} & z_{m,2} & \cdots & z_{m,n} & z_{m,n+1} \\ z_{m+1,1} & z_{m+1,2} & \cdots & z_{m+1,n} & z_{m+1,n+1} \end{bmatrix}. \tag{1}$$

For notational convenience, let $\boldsymbol{Z} = [\boldsymbol{X}, \boldsymbol{y}]$ where $\boldsymbol{X}$ denotes the first n column of $\boldsymbol{Z}$ and $\boldsymbol{y}$ denotes the last column of $\boldsymbol{Z}$. Without loss of generality, we assume the target gene with a missing value is the $(n+1)$-th gene. Let $\boldsymbol{y} = [y_1, \dots, y_m, y_{m+1}]^{\mathrm{T}}$ denote its expression profiles with y_{m+1} as the missing value. The other n genes in the other m experiments $\boldsymbol{X}$ are then used to find the similar genes to the $(n+1)$-th gene y. The data $\boldsymbol{X}$ from the first m experiments are used to select genes among the n genes that are highly correlated with the target gene y. The following linear regression model is used to relate the gene expression levels of the target gene and other genes:

$$y_i = \boldsymbol{X}_i \boldsymbol{\beta} + e_i, \quad i = 1, \dots, m, \tag{2}$$

where $\boldsymbol{X}_i$ is the i-th row of the matrix $\boldsymbol{X}, \boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_n)^{\mathrm{T}}$ is the vector of regression parameters and the i.i.d. noise e_i follows $e_i \sim \mathcal{N}(0, \sigma^2)$. Note that $\boldsymbol{\beta}$ is fixed over all m experiments. Since typically n is large, to make an accurate prediction, we must find a set of genes that is highly correlated with y. The $(m+1)$-th experiment of the most relevant genes in $\boldsymbol{X}$ are then used to predict the value of the $(n+1)$-th gene in the $(m+1)$-th experiment, i.e. y_{m+1}.

2.2 Bayesian gene selection

Define $\boldsymbol{\gamma}$ as the $n \times 1$ vector of indicator variables γ_j such that $\gamma_j = 0$ if $\beta_j = 0$ (the variable is not selected) and $\gamma_j = 1$ if $\beta_j \neq 0$ (the variable is selected). Given $\boldsymbol{\gamma}$, let $\boldsymbol{\beta_\gamma}$ consist of all non-zero elements of $\boldsymbol{\beta}$ and let $\boldsymbol{X_\gamma}$ be the columns of $\boldsymbol{X}$ corresponding to those of $\boldsymbol{\gamma}$ that are equal to 1.

To treat gene selection under the Bayesian framework, we make the following assumptions on the priors of the parameters in (2). Firstly, given $\boldsymbol{\gamma}$ and σ^2, the prior for $\boldsymbol{\beta_\gamma}$ is

$$\boldsymbol{\beta_\gamma} \sim \mathcal{N}\left[0, c\sigma^2(\boldsymbol{X}_{\boldsymbol{\gamma}}^{\mathrm{T}} \boldsymbol{X_\gamma})^{-1}\right],$$

where we empirically set $c = 100$ (Albert *et al.*, 1993; Smith and Kohn, 1997; Lee *et al.*, 2003). Given $\boldsymbol{\gamma}$, the prior for σ^2 is assumed to be a conjugate inverse-Gamma distribution, $p(\sigma^2|\boldsymbol{\gamma}) \propto \mathcal{IG}(v_0/2, \nu_0/2)$. When $v_0 = 0$ and $\nu_0 = 0$, we obtain Jeffrey's uninformative prior, i.e. $p(\sigma^2) \propto 1/\sigma^2$. Bayesian gene selection using a binomial probit regression model is discussed by Lee *et al.* (2003), where it is assumed that $\sigma^2 = 1$. Moreover, $\{\gamma_j\}_{j=1}^n$ are assumed to be independent with $p(\gamma_j = 1) = \pi_j, j = 1, \dots, n$, where π_j is the probability to select gene j. Obviously, if we want to select 10 genes from all n genes, then π_j may be set as $10/n$. In this paper we empirically set $\pi_j = 15/n$ for all genes, based on the total sample number $m = 22$. If π_j is chosen to take a larger value, then we found that often times $(\boldsymbol{X}_{\boldsymbol{\gamma}}^{\mathrm{T}} \boldsymbol{X_\gamma})^{-1}$ is singular.

Here, we introduce the Bayesian variable selection principle (Smith and Kohn, 1997). A Gibbs sampler is employed to estimate the parameters. Denote

$$S(\boldsymbol{\gamma}, \boldsymbol{y}) \triangleq \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - \frac{c}{c+1}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{X_\gamma}(\boldsymbol{X}_{\boldsymbol{\gamma}}^{\mathrm{T}}\boldsymbol{X_\gamma})^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^{\mathrm{T}}\boldsymbol{y}, \tag{3}$$

where $\boldsymbol{y} = [y_1, y_2, \dots, y_m]^{\mathrm{T}}$. Define $n_\gamma = \sum_{i=1}^n \gamma_i$. In the Appendix on the associated web site (also see Smith and Kohn, 1997). We show that

$$p(\boldsymbol{y}|\boldsymbol{\gamma}) \propto \int_\sigma \left\{ \int_{\beta_\gamma} p(\boldsymbol{y}|\boldsymbol{\beta_\gamma}, \sigma^2) p(\boldsymbol{\beta_\gamma}|\sigma^2) \mathrm{d}p(\boldsymbol{\beta_\gamma}) \right\} p(\sigma^2)\, \mathrm{d}\sigma^2$$
$$\propto (1+c)^{-n_\gamma/2} S(\boldsymbol{\gamma}, \boldsymbol{y})^{-m/2}. \tag{4}$$

Then the posterior distribution of $\boldsymbol{\gamma}$ is

$$p(\boldsymbol{\gamma}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})$$
$$\propto (1+c)^{-n_\gamma/2} S(\boldsymbol{\gamma}, \boldsymbol{y})^{-m/2} \prod_{j=1}^n \pi_j^{\gamma_j}(1-\pi_j)^{1-\gamma_j}. \tag{5}$$

In the Appendix, we show that the posterior distributions of σ^2 and $\boldsymbol{\beta}$ are given respectively by

$$p(\sigma^2|\boldsymbol{y}, \boldsymbol{X}_\gamma) \propto \mathcal{IG}\left(\frac{m}{2}, \frac{S(\boldsymbol{\gamma}, \boldsymbol{y})}{2}\right), \tag{6}$$

$$p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}_\gamma, \sigma^2) \propto \mathcal{N}(\boldsymbol{V_\gamma} \boldsymbol{X}_{\boldsymbol{\gamma}}^{\mathrm{T}} \boldsymbol{y}, \sigma^2 \boldsymbol{V_\gamma}), \tag{7}$$

where

$$\boldsymbol{V_\gamma} \triangleq \frac{c}{1+c}(\boldsymbol{X}_{\boldsymbol{\gamma}}^{\mathrm{T}} \boldsymbol{X_\gamma})^{-1}. \tag{8}$$

Finally, the Gibbs sampling algorithm for jointly estimating $\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma^2$ is as follows:

1. Draw $\boldsymbol{\gamma}$ from $p(\boldsymbol{\gamma}|\boldsymbol{y})$ in (5). In fact, we sample each γ_j independently from

$$p(\gamma_j|\boldsymbol{y}, \gamma_{i\neq j}) \propto (1+c)^{-(n_\gamma/2)} \exp\left[-\frac{1}{2}S(\boldsymbol{\gamma}, \boldsymbol{y})\right] \times \pi_j^{\gamma_j}(1-\pi_j)^{1-\gamma_j}, \quad j = 1, \ldots, n. \tag{9}$$

2. Draw σ^2 from $p(\sigma^2|\boldsymbol{y}, \boldsymbol{\gamma})$ in (6).
3. Draw $\boldsymbol{\beta}$ from $p(\boldsymbol{\beta}|\boldsymbol{\gamma}, \boldsymbol{y})$ in (7).

In this study, the initial parameters are randomly set. $T = 35\ 000$ iterations are implemented with the first 5000 as the burn-in period to obtain the Monte Carlo samples $\{\boldsymbol{\gamma}^{(t)}, \sigma^{2^{(t)}}, \boldsymbol{\beta}^{(t)}, t = 1, \ldots, T\}$. We count the number of times that each gene appears in $\{\boldsymbol{\gamma}^{(t)}, t = 5001, \ldots, T\}$. The genes with the highest appearance frequencies play the strongest role in predicting the target gene.

2.3 Missing-value prediction using the strongest genes

Now assume the genes corresponding to the non-zero elements of $\boldsymbol{\gamma}$ are the strongest obtained by the Bayesian variable selection algorithm. Let $\boldsymbol{X}_{m+1,\gamma}$ denote the $(m+1)$-th expression profiles of these strongest genes. There are three methods to estimate $\boldsymbol{\beta}_\gamma$ and predict the missing value y_{m+1}. One is to just use least-squares, i.e. $\boldsymbol{\beta}_\gamma = (\boldsymbol{X}_\gamma^{\mathrm{T}}\boldsymbol{X}_\gamma)^{-1}\boldsymbol{X}_\gamma^{\mathrm{T}}\boldsymbol{y}$. Then y_{m+1} is estimated by $\hat{y}_{m+1} = \boldsymbol{X}_{m+1}\boldsymbol{\beta}_\gamma$. A second is to adopt model averaging in the gene selection step to get $\boldsymbol{\beta}$. However, since during gene selection the number of genes selected varies from one Gibbs iteration to another, averaging the values of $\boldsymbol{\beta}$ corresponding to different models is problematic. We adopt the following method. For fixed $\boldsymbol{\gamma}$, we again use a Gibbs sampler to estimate the linear regression coefficients $\boldsymbol{\beta}$ as follows: first draw $\boldsymbol{\beta}_\gamma$ according to (7), then draw σ^2 according to (6) and iterate the two steps. $\tilde{T} = 1500$ iterations are implemented with the first 500 as the burn-in to obtain the Monte Carlo samples $\{\tilde{\boldsymbol{\beta}}^{(t)}, \tilde{\sigma}^{2^{(t)}}, t = 501, \ldots, \tilde{T}\}$. The missing value y_{m+1} is estimated by

$$\hat{y}_{m+1} = \frac{1}{\tilde{T}} \sum_{t=501}^{\tilde{T}} \boldsymbol{X}_{m+1,\gamma} \tilde{\boldsymbol{\beta}}_\gamma^{(t)}.$$

Note that if no prior is applied to $\boldsymbol{\beta}$, we can use the least-squares method. We next define the normalized RMS error for the predictor. Assume the missing values are y_{ij} and the corresponding estimates are $\hat{y}_{ij}, i = 1, \ldots, N_i, j = 1, \ldots, N_j$. Then the normalized RMS is

$$RMS \triangleq \sqrt{\frac{1}{N_i \cdot N_j} \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} (y_{ij} - \hat{y}_{ij})^2}.$$

3 FAST IMPLEMENTATION ISSUES

The computational complexity of the Bayesian variable selection algorithm is high. For example, if there are 3000 gene variables, then for each iteration we have to compute the inverse $(\boldsymbol{X}_{\boldsymbol{\gamma}}^{\mathrm{T}}\boldsymbol{X_\gamma})^{-1}$ 3000 times because we need to sample γ_j for each gene according to (9). Our concerns about computational complexity are mitigated by the fact that missing-value estimation need only be done once for a set of experiments and could be implemented on a supercomputer; nevertheless, it is possible to apply some procedures to speed up the computation while still achieving good results. In fact, all experimental results discussed in this paper have been obtained using the computational speed-ups discussed in this section.

3.1 Pre-selection

The pre-selection method selects genes with expression profiles similar to the target gene. If we consider gene y that has one missing value in experiment $m+1$, then the pre-selection procedure finds u other genes, which have values in experiment $m+1$, with expression most similar to y in experiments 1 to m in the Euclidean distance sense. In this paper, we set $u = 200$.

3.2 Computation of $p(\gamma_j|\boldsymbol{y}, \gamma_{i\neq j})$ in (9)

Because γ_j only takes 0 or 1, we can take a close look at $p(\gamma_j = 1|\boldsymbol{y}, i \neq j)$ and $p(\gamma_j = 0|\boldsymbol{y}, i \neq j)$. Let $\boldsymbol{\gamma}^1 = (\gamma_1, \ldots, \gamma_{j-1}, \gamma_j = 1, \gamma_{j+1}, \ldots, \gamma_n)$ and $\boldsymbol{\gamma}^0 = (\gamma_1, \ldots, \gamma_{j-1}, \gamma_j = 0, \gamma_{j+1}, \ldots, \gamma_n)$. After straightforward computation of (9), we have

$$p(\gamma_j = 1|\boldsymbol{y}, \gamma_{i\neq j}) \propto \frac{1}{1+h}, \tag{10}$$

with

$$h = \frac{1-\pi_j}{\pi_j}\left(\frac{S(\boldsymbol{\gamma}^1, \boldsymbol{y})}{S(\boldsymbol{\gamma}^0, \boldsymbol{y})}\right)^{m/2}\sqrt{1+c}. \tag{11}$$

If $\boldsymbol{\gamma} = \boldsymbol{\gamma}^0$ before γ_j is generated, that means we have obtained $S(\boldsymbol{\gamma}^0, \boldsymbol{y})$, then we only need to compute $S(\boldsymbol{\gamma}^1, \boldsymbol{y})$, and vice versa.

3.3 Fast computation of $S(\boldsymbol{\gamma}, \boldsymbol{y})$ in (3)

The key is to compute $S(\boldsymbol{\gamma}, \boldsymbol{y})$ fast when a gene variable is added or removed from $\boldsymbol{\gamma}$. Denote

$$E(\boldsymbol{\gamma}, \boldsymbol{y}) \triangleq \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}}\boldsymbol{X}_{\boldsymbol{\gamma}}(\boldsymbol{X}_{\boldsymbol{\gamma}}^{\mathrm{T}}\boldsymbol{X}_{\boldsymbol{\gamma}})^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^{\mathrm{T}}\boldsymbol{y}. \tag{12}$$

This can be computed using the fast QR decomposition, QR-delete and QR-insert algorithms when a variable is added or removed (Seber, 1984, Ch. 10.1.1b). Now we want to estimate $S(\boldsymbol{\gamma}, \boldsymbol{y})$ in (3). Comparing (12) and (3), one can obtain the following equation:

$$\boldsymbol{y}^{\mathrm{T}}\boldsymbol{X}_{\boldsymbol{\gamma}}(\boldsymbol{X}_{\boldsymbol{\gamma}}^{\mathrm{T}}\boldsymbol{X}_{\boldsymbol{\gamma}})^{-1}\boldsymbol{X}_{\boldsymbol{\gamma}}^{\mathrm{T}}\boldsymbol{y} = [S(\boldsymbol{\gamma}, \boldsymbol{y}) - E(\boldsymbol{\gamma}, \boldsymbol{y})](c+1). \tag{13}$$

Substituting (13) to (3), after straightforward computation, $S(\boldsymbol{\gamma}, \boldsymbol{y})$ is given by

$$S(\boldsymbol{\gamma}, \boldsymbol{y}) = \frac{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} + cE(\boldsymbol{\gamma}, \boldsymbol{y})}{1+c}. \tag{14}$$

Thus after computing $E(\boldsymbol{\gamma}, \boldsymbol{y})$ using QR decomposition, QR-delete or QR-insert algorithms, we then can obtain $S(\boldsymbol{\gamma}, \boldsymbol{y})$. We summarize our fast Bayesian variable selection algorithm as follows.

ALGORITHM 1: Fast Bayesian variable selection algorithm

```
• Pre-select genes based on the K-nearest neighbor distance;
• Initialization: Randomly set initial parameters $\boldsymbol{\gamma}^{(0)}$, $\sigma^{2^{(0)}}$, $\boldsymbol{\beta}^{(0)}$;
• For $t = 1, \ldots, 35\,000$
  — Draw $\boldsymbol{\gamma}^{(t)}$. For $j = 1, \ldots, n$
      ∗ Compute $S(\boldsymbol{\gamma}^{(t)}, \boldsymbol{y})$ using QR-delete or QR-insert;
      ∗ Compute $p(\gamma_j = 1|\boldsymbol{y}, \gamma_{i\neq j}^{(t)})$ according to (11);
      ∗ Draw $\gamma_j^{(t)}$ from $p(\gamma_j = 1|\boldsymbol{y}, \gamma_{i\neq j}^{(t)})$.
  — Draw $\sigma^{2^{(t)}}$ according to (6).
  — Draw $\boldsymbol{\beta}^{(t)}$ according to (7).
• Endfor
• Count the frequency of each gene appeared in $\boldsymbol{\gamma}^{(t)}, t = 5001, \ldots, 35\,000$.
```

At each iteration, the number of selected genes depends on the sampling outcome of γ_j. Although we set $\pi_j = 15/n$ empirically, we cannot avoid the case that the number of selected genes is bigger than the sample size m. If this happens, we need to remove this case because $(\boldsymbol{X}_{\boldsymbol{\gamma}}^{\mathrm{T}}\boldsymbol{X}_{\boldsymbol{\gamma}})^{-1}$ does not exist. The above algorithm is for single missing-value estimation. When there are multiple missing values, the algorithm should be applied to estimate each missing value.

4 NONLINEAR REGRESSION WITH BAYESIAN GENE SELECTION

In Zhou *et al.* (2003), we have found some genes show a strong nonlinear property, so here we also discuss the nonlinear regression missing-value estimation. The problem is the same as stated in Section 2. We denote $\boldsymbol{y} = [y_1, \ldots, y_m]^{\mathrm{T}}$, $\boldsymbol{X}_i = [x_{i1}, \ldots, x_{in}]$ for $i = 1, \ldots, m$, and $\boldsymbol{x} = [x_1, \ldots, x_n]^{\mathrm{T}}$. We use a nonlinear regression model composed of a linear term plus a nonlinear term given by

$$y = \sum_{i=1}^{n}\alpha_i x_i + \sum_{k=1}^{\kappa}\beta_k\phi_k(x_1, \ldots, x_n) + e, \tag{15}$$

with

$$\phi_k(x_1, \ldots, x_n) \triangleq \exp\{-\lambda_k\|\boldsymbol{x} - \boldsymbol{\mu}_k\|\}, \quad k = 1, \ldots, \kappa, \tag{16}$$

where $\|\cdot\|$ is the Euclidean norm; $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]^{\mathrm{T}}$, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_\kappa]^{\mathrm{T}}$ are the vectors of regression parameters; the additive noise term e follows a normal distribution, i.e. $e \sim \mathcal{N}(0, \sigma^2)$; $\{\boldsymbol{\mu}_k\}_{k=1}^{\kappa}$ are the centers of the κ clusters obtained by using fuzzy-c means clustering; and the parameters $\{\lambda_k\}_{k=1}^{\kappa}$ are chosen empirically. Here, we set $\kappa = 2$ and $\lambda_k = 2,\ k = 1, \ldots, \kappa$, and have

$$\boldsymbol{y} = \boldsymbol{Z}_1\boldsymbol{\alpha} + \boldsymbol{Z}_2\boldsymbol{\beta} + \boldsymbol{e}, \tag{17}$$

where $\boldsymbol{e} = [e_1, e_2, \ldots, e_m]^{\mathrm{T}}$,$\boldsymbol{Z}_1 = [\boldsymbol{X}_1^{\mathrm{T}}, \ldots, \boldsymbol{X}_m^{\mathrm{T}}]^{\mathrm{T}}$, and

$$\boldsymbol{Z}_2 = \begin{bmatrix} \phi_1(\boldsymbol{X}_1) & \cdots & \phi_\kappa(\boldsymbol{X}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\boldsymbol{X}_m) & \cdots & \phi_\kappa(\boldsymbol{X}_m) \end{bmatrix}.$$

Note that (17) can be further written as

$$\boldsymbol{y} = \hat{\boldsymbol{X}}\hat{\boldsymbol{\alpha}} + \boldsymbol{e}, \tag{18}$$

where

$$\hat{\boldsymbol{X}} \triangleq [\boldsymbol{Z}_1, \boldsymbol{Z}_2]$$

$$= \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} & \phi_1(\boldsymbol{X}_1) & \cdots & \phi_\kappa(\boldsymbol{X}_1) \\ x_{21} & x_{22} & \cdots & x_{2n} & \phi_1(\boldsymbol{X}_2) & \cdots & \phi_\kappa(\boldsymbol{X}_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} & \phi_1(\boldsymbol{X}_m) & \cdots & \phi_\kappa(\boldsymbol{X}_m) \end{bmatrix}. \tag{19}$$

and $\hat{\boldsymbol{\alpha}} \triangleq [\boldsymbol{\alpha}^{\mathrm{T}}, \boldsymbol{\beta}^{\mathrm{T}}]^{\mathrm{T}}$.

Now we apply the same gene selection algorithm and missing-value estimation algorithm as discussed in Sections 2 and 3 to (18). Although the problem is nonlinear in terms of $\boldsymbol{X}$ in this nonlinear case, it is linear in terms of $\phi(\boldsymbol{X})$, so the same formula can be used. Note that we can estimate the parameters $\boldsymbol{\mu}_k, k = 1, \ldots, \kappa$ in (16) using an MCMC technique similar to the approach in (Zhou *et al.*, 2003) if we want to obtain better estimation performance at the expense of a significant increase in computational complexity.

5 EXPERIMENTAL RESULTS

We consider hereditary breast cancer data (Hedenfalk *et al.*, 2001). Application to a small, round blue-cell tumor (Khan *et al.*, 2001) data set is given on the associated web site. Considering the high computational complexity of the new methods, we assess the performance of the KNNimpute, linear regression and nonlinear regression methods from three aspects: number of selected genes used for different methods (from 1 to 18 genes for the breast cancer data; and 1 to 20 genes for SRBCT data); comparison of the three methods based on estimation performance on different amounts of missing data, from 1 to 5%; distribution of errors for the three methods for fixed $K = 7$ at 1% of data missing.

In Hedenfalk *et al.* (2001), cDNA microarrays were used in conjunction with classification algorithms to show the feasibility of using differences in global gene expression profiles to separate BRCA1 and BRCA2 mutation-positive breast cancers. The hereditary breast cancer data can be downloaded from the original author's web page (Hedenfalk *et al.*, 2001). Twenty-two breast tumor samples from 21 breast cancer patients were examined: seven BRCA1, eight BRCA2 and seven sporadic. 3226 genes were used for each tumor sample. We test algorithm performance using the natural log of the breast cancer ratio data over different values of K. Some percentage of data is randomly deleted and then estimated by each missing-value algorithm. The results are the average of 50 experiments using cross-validation.

Figure 1 shows the normalized RMS errors when using the linear and nonlinear regression methods, and the KNNimpute procedure over 1 and 5% data missing. The nonlinear regression method performs best over the range of K, and the linear regression method preforms slightly poorer. Both significantly outperform the KNNimpute method. The performances of the linear regression and nonlinear regression predictors degrade more quickly when the number of selected genes exceeds 14. This degradation results from several reasons: $(\boldsymbol{X}_\gamma^{\mathrm{T}}\boldsymbol{X}_\gamma)^{-1}$ gradually becoming singular, greater difficulty to estimate the model parameters when gene variables are increased; and preselection of 200 genes to speed-up the regression algorithms (which can be mitigated by employing less speed-up). In fact, degradation for increasing K is not important because all three algorithms achieve close to their best performances in a large range of K, and in practice we would use a K from that range, say $K = 7$, to obtain good performance while keeping K small for computational purposes and centered within the range of good performance for robustness.

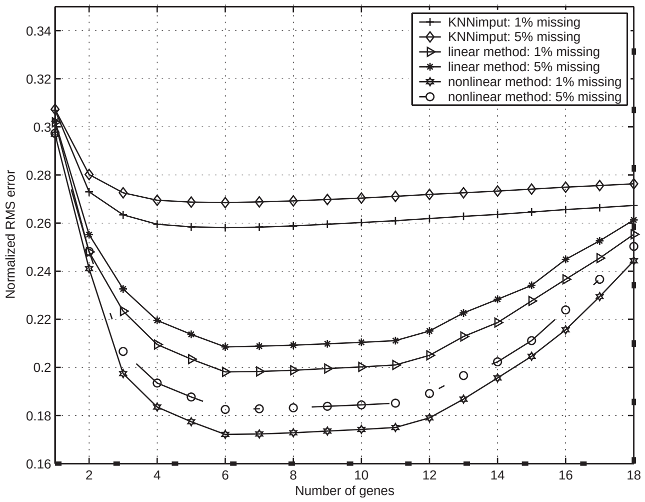

Fig. 1. Effect of the number selected genes used for different methods in Experiment 1.

Further illustration of the improved accuracy of the linear and nonlinear regression methods is given in Figure 2, which shows the histograms for the distribution of normalized RMS errors for the three methods. The two new algorithms (as well as KNNimpute) are robust relative to increasing the percentage of missing values, This is shown in Figure 3 with the percentage of missing values between 1 and 5% and $K = 7$.

6 CONCLUSION

This paper proposes two new methods for missing-value estimation: linear and nonlinear regression with Bayesian gene selection. We have analyzed their performance on data from hereditary breast cancer and from small round blue-cell tumors. The results show that the linear and nonlinear approaches with Bayesian gene selection compare favorably with the KNNimpute method in terms of the normalized RMS error. This comparison has added significance since, as shown

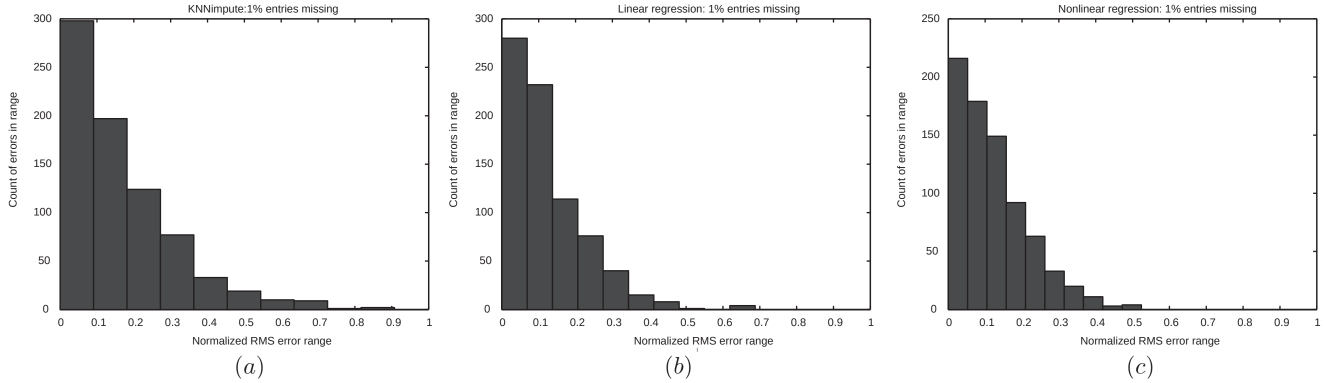

Fig. 2. Error histograms of different estimation methods and 1% data missing rate: (**a**) the KNNimpute, (**b**) the linear regression and (**c**) the nonlinear regression.

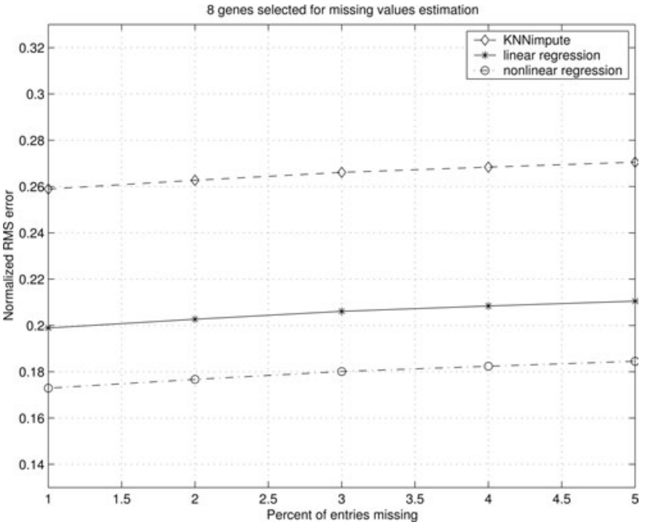

Fig. 3. Performance comparison of the KNNimpute, the linear and nonlinear regression methods under different different data missing percentages in Experiment 1.

by Troyanskaya *et al*. (2001), KNNimpute compares favorably with filling with zeros and row average, as well as SVDimpute.

ACKNOWLEDGEMENTS

The authors thank the reviewers for their careful reading of our manuscript and their constructive comments. This research was supported by the National Human Genome Research Institute. X.W. was supported in part by the US National Science Foundation (DMS-0225692)

REFERENCES

Albert,J. and Chib,S. (1993) Bayesian analysis of binary and polychotomous response data. *J. Am. Stat. Assoc.*, **88**, 669–679.

Alizadeh,A.A., Eisen,M.B., Davis,R.E., Ma,C., Lossos,I.S., Rosenwald,A., Boldrick,J.C., Sabet,H., Tran,T., Yu,X. *et al*. (2000) Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, **403**, 503–511.

Arbeitman,M.N., Furlong,E.E.M., Imam,F., Johnson,E., Null,B.H., Baker,B.S., Krasnow,M.A., Scott,M.P., Davis,R.W. and White,K.P. (2002) Gene expression during the life cycle of *Drosophila melanogaster*. *Science*, **297**, 2270–2275.

Butte,A.J., Ye,J., Niederfellner,G., Rett,K., Hring,H., White,M.F. and Kohane,I.S. (2001) Determining significant fold differences in gene expression analysis. *Pac. Symp. Biocomput.*, **6**, 6–17.

Chen,Y., Kamat,V., Dougherty,E.R., Bittner,M., Meltzer,P.S. and Trent,J.M. (2002) Ratio statistics of gene expression levels and applications to microarray data analysis. *Bioinformatics*, **18**, 1207–1215.

Hedenfalk,I., Duggan,D., Chen,Y., Radmacher,M., Bittner,M., Simon,R., Meltzer,P., Gusterson,B., Esteller,M., Raffeld,M. *et al*. (2001) Gene expression profiles in hereditary breast cancer. *N. Engl. J. Med.*, **344**, 539–548.

Khan,J., Wei,J.S., Ringner,M., Saal,L.H., Ladanyi,M., Westermann,F., Berthold,F., Schwab,M., Antonescu,C.R., Peterson,C. and Meltzer,P.S. (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat. Med.*, **7**, 673–679.

Lee,K.E., Sha,N., Dougherty,E.R., Vannucci,M. and Mallick,B.K. (2003) Gene selection: a Bayesian variable selection approach. *Bioinformatics*, **19**, 90–97.

Smith,M. and Kohn,R. (1997) Nonparametric regression using Bayesian variable selection. *J. Econometr.*, **75**, 317–344.

Seber,G.A.F. (1984) *Multivariate Observatoins*. Wiley, New York.

Troyanskaya,O., Cantor,M., Sherlock,G., Brown,P., Hastie,T., Tibshirani,R., Botstein,D. and Altman,R.B. (2001) Missing value estimation methods for DNA microarrays. *Bioinformatics*, **17**, 520–525.

Zhou,X., Wang,X. and Dougherty,E.R. (2003) Construction of genomic networks using mutual-information clustering and reversible-jump Markov-chain-Monte-Carlo predictor design. *Signal Processing*, **84**, 745–761.