EURASIP Journal on Applied Signal Processing 2004:1, 115–124

Gene Prediction Using Multinomial Probit Regression with Bayesian Gene Selection

Xiaobo Zhou
Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, USA
Email: zxb@ee.tamu.edu

Xiaodong Wang
Department of Electrical Engineering, Columbia University, New York, NY 10027, USA
Email: wangx@ee.columbia.edu

Edward R. Dougherty
Department of Electrical Engineering, Texas A&M University, 3128 TAMU College Station, TX 77843-3128, USA
Department of Pathology, University of Texas MD Anderson Cancer Center, Houstan, TX 77030, USA
Email: e-dougherty@tamu.edu

Received 3 April 2003; Revised 1 September 2003

A critical issue for the construction of genetic regulatory networks is the identification of network topology from data. In the context of deterministic and probabilistic Boolean networks, as well as their extension to multilevel quantization, this issue is related to the more general problem of expression prediction in which we want to find small subsets of genes to be used as predictors of target genes. Given some maximum number of predictors to be used, a full search of all possible predictor sets is combinatorially prohibitive except for small predictors sets, and even then, may require supercomputing. Hence, suboptimal approaches to finding predictor sets and network topologies are desirable. This paper considers Bayesian variable selection for prediction using a multinomial probit regression model with data augmentation to turn the multinomial problem into a sequence of smoothing problems. There are multiple regression equations and we want to select the same strongest genes for all regression equations to constitute a target predictor set or, in the context of a genetic network, the dependency set for the target. The probit regressor is approximated as a linear combination of the genes and a Gibbs sampler is employed to find the strongest genes. Numerical techniques to speed up the computation are discussed. After finding the strongest genes, we predict the target gene based on the strongest genes, with the coefficient of determination being used to measure predictor accuracy. Using malignant melanoma microarray data, we compare two predictor models, the estimated probit regressors themselves and the optimal full-logic predictor based on the selected strongest genes, and we compare these to optimal prediction without feature selection.

Keywords and phrases: gene microarray, multinomial probit regression, Bayesian gene selection, genetic regulatory networks.

1. INTRODUCTION

The advent of high throughput gene expression microarray technology has stimulated the development of mathematical models for genetic regulatory networks, in particular, discrete models such as Bayesian networks [1, 2, 3, 4], Boolean networks [5, 6, 7, 8], probabilistic Boolean networks [9, 10], and the generalization of both deterministic and probabilistic Boolean networks to multilevel quantization [11, 12]. A critical issue for network construction is the identification of network topology from the data. This issue is related to the more general problem of expression prediction in which we want to find small subsets of genes to be used as predictors of target genes [11, 13]. Given some maximum number of predictors to be used, ideally one would like to search over all possible predictor sets to find those that are the best relative to some measure of prediction such as the coefficient of determination [14]; however, such a search is combinatorially prohibitive except for small predictors sets, and even then, may require supercomputing [15]. Consequently, this has lead to an effort to find other, perhaps suboptimal, approaches to finding predictor sets, and the concomitant network topologies. Two such efforts involve minimum description length [16], mutual-information-based clustering [12], and incremental inclusion of predictor variables [17].

The search for good predictor sets is a form of feature reduction, which in the context of expression-based classification involves methods to reduce the set of genes from which

good feature sets can be formed. Owing to the importance of classification and the extremely large number of genes from which to form classifiers from microarray data, several methods have been proposed, including the support vector machine method [18], minimum description length [19], voting [20], and Bayesian variable selection [21, 22].

In this paper, we focus on Bayesian variable selection for prediction using a multinomial regression model (probit regressor) with data augmentation to turn the multinomial problem into a sequence of smoothing problems [23]. In a sense, this work extends the method of [22], except that here the input and output values are ternary instead of analog and binary, respectively. This means that there are multiple regression equations and we want to select the same strongest genes for all regression equations to constitute a target predictor set or, in the context of a genetic regulatory network, the dependency set for the target. The probit regressor is approximated as a linear combination of the genes and a Gibbs sampler is employed to find the strongest genes. Since this method has high computational complexity, we discuss some numerical techniques to speed up the computation. After finding the strongest genes, we predict the target gene based on the strongest genes, with the coefficient of determination being used to measure predictor accuracy. Normally, when trying to identify network topologies and related problems, one uses time series data. In this paper, we aim at the same goal using static data, that is, malignant melanoma microarray data [24]. Using malignant melanoma microarray data, we compare two predictor models: (1) the estimated probit regressors themselves and (2) the optimal full-logic predictor based on the selected strongest genes. As must be the case, full-logic prediction with the strongest genes will outperform the regressor model with the strongest genes; nevertheless, the fundamental issue in this paper is feature reduction and this is accomplished satisfactorily if the optimal full-logic predictor performs well with the selected feature set.

2. MULTINOMIAL PROBIT REGRESSION WITH BAYESIAN GENE SELECTION

2.1. Problem formulation

Assume that there are $n+1$ genes, say, $x_1,\ldots,x_n,x_{n+1}$. Without loss of generality, we assume that the target gene is x_{n+1}, and let w denote this target gene. Then $\mathbf{w} = [w_1,\ldots,w_m]^T$ denotes the normalized expression profiles of the target gene (e.g., for the normalized ternary expression data, $w_j = 1$ indicates that the sample j is up-regulated; $w_j = -1$ indicates that the sample j is down-regulated; and $w_j = 0$ indicates that the sample j is invariant). Denote

$$\mathbf{X} = \begin{bmatrix} \text{Gene 1} & \text{Gene 2} & \cdots & \text{Gene } n \\ x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \tag{1}$$

as the normalized expression profiles of genes $x_1,\ldots,x_n$. The gene selection problem is to find some genes from $x_1,\ldots,x_n$ that are useful in predicting some target gene w. Here, we consider a more general case of gene prediction, that is, assume that the gene expression profiles are normalized to K levels.

The perceptron has been proved to be an effective model to model the relationship between the target gene and the other genes [25]. Here, we study this problem by using probit regression with Bayesian gene selection. Let $\mathbf{X}_i$ denote the ith row of matrix $\mathbf{X}$ in (1). In the binomial probit regression, that is, when $K = 2$, the relationship between w_i and the gene expression levels $\mathbf{X}_i$ is modeled as a probit regressor [23] which yields

$$P(w_i = 1|\mathbf{X}_i) = \Phi(\mathbf{X}_i\boldsymbol{\beta}), \quad i = 1,\ldots,m, \tag{2}$$

where $\boldsymbol{\beta} = (\beta_1,\beta_2,\ldots,\beta_n)^T$ is the vector of regression parameters and Φ is the standard normal cumulative distribution function. Introduce m independent latent variable $z_1,\ldots,z_m$, where $z_i \sim N(\mathbf{X}_i\boldsymbol{\beta},1)$, that is,

$$z_i = \mathbf{X}_i\boldsymbol{\beta} + e_i, \quad i = 1,\ldots,m, \tag{3}$$

and $e_i \sim N(0,1)$. Define $\boldsymbol{\gamma}$ as the $n\times 1$ indicator vector with the jth element 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). The Bayesian variable selection is to estimate $\boldsymbol{\gamma}$ from the posteriori distribution $p(\boldsymbol{\gamma}|\mathbf{z})$. See [11] for details.

However, when $K > 2$, the situation is different from the binomial case because we have to construct $K-1$ regression equations similar to (3). Introduce $K-1$ latent variables $z_1,\ldots,z_{K-1}$ and $K-1$ regression equations such that $z_k = \mathbf{X}\boldsymbol{\beta}_k + e_k$, $k = 1,\ldots,K-1$, where $e_k \sim N(0,1)$. Let z_k take m values $\{z_{k,1},\ldots,z_{k,m}\}$. Using matrix form, it can be further written as

$$\begin{aligned} z_{k,1} &= \mathbf{X}_1\boldsymbol{\beta}_k + e_{k,1}, \\ z_{k,2} &= \mathbf{X}_2\boldsymbol{\beta}_k + e_{k,2}, \\ &\vdots \\ z_{k,m} &= \mathbf{X}_m\boldsymbol{\beta}_k + e_{k,m}, \end{aligned} \tag{4}$$

where $k = 1,\ldots,K-1$. Denote $\mathbf{z}_k \triangleq [z_{k,1},\ldots,z_{k,m}]^T$ and $\mathbf{e}_k \triangleq [e_{k,1},\ldots,e_{k,m}]^T$. Then (4) can be rewritten as

$$\mathbf{z}_k = \mathbf{X}\boldsymbol{\beta}_k + \mathbf{e}_k, \quad k = 1,\ldots,K-1. \tag{5}$$

This model is called the multinomial probit model. For background on multinomial probit models, see [26]. Note that we do not have the observations of $\{\mathbf{z}_k\}_{k=1}^{K-1}$, which makes it difficult to estimate the parameters in (5).

Here, we discuss how to select the same strongest genes for the different regression equations. The model is a little different from (5), that is, the selected genes do not change with the different regression equations. Note that the

(i) Draw $\boldsymbol{\gamma}$ from $p(\boldsymbol{\gamma}|\mathbf{z}_1,\ldots,\mathbf{z}_{K-1})$. We usually sample each γ_i independently from

$$\begin{aligned} &p(\gamma_i|\mathbf{z}_1,\ldots,\mathbf{z}_{K-1},\gamma_{j\neq i}) \\ &\quad\propto p(\mathbf{z}_1,\ldots,\mathbf{z}_{K-1}|\boldsymbol{\gamma})\,p(\gamma_i) \\ &\quad\propto (1+c)^{-(K-1)n_\gamma/2}\exp\left\{-\frac{1}{2}\sum_{k=1}^{K-1}S(\boldsymbol{\gamma},\mathbf{z}_k)\right\}\pi_i^{\gamma_i}(1-\pi_i)^{1-\gamma_i}, \end{aligned} \tag{10}$$

$n_\gamma=\sum_{j=1}^{n}\gamma_j$, $c=10$, and $\pi_i=P(\gamma_i=1)$ are prior probabilities to select the jth gene. It is set as $\pi_i=8/n$ according to the very small sample size. If π_i takes a larger value, we find oftentimes that $(\mathbf{X}_\gamma{}^T\mathbf{X}_\gamma)^{-1}$ does not exist.

(ii) Draw $\boldsymbol{\beta}_k$ from

$$p(\boldsymbol{\beta}_k|\boldsymbol{\gamma},\mathbf{z}_k)\propto\mathcal{N}(\mathbf{V}_\gamma\mathbf{X}_\gamma{}^T\mathbf{z}_k,\mathbf{V}_\gamma), \tag{11}$$

where $\mathbf{V}_\gamma=(c/(1+c))(\mathbf{X}_\gamma{}^T\mathbf{X}_\gamma)^{-1}$.

(iii) Draw $\mathbf{z}_k=[z_{k,1},\ldots,z_{k,m}]^T$, $k=1,\ldots,K$, from a truncated normal distribution as follows [27].

For $i=1,2,\ldots,m$

If $w_i=k$, then draw $z_{k,i}$ according to $z_{k,i}\sim N(\mathbf{X}_\gamma\boldsymbol{\beta}_k,1)$ truncated left by $\max_{j\neq k}z_{j,i}$, that is,

$$z_{k,i}\sim\mathcal{N}(\mathbf{X}_\gamma\boldsymbol{\beta}_k,1)1_{\{z_{k,i}>\max_{j\neq k}z_{j,i}\}}. \tag{12}$$

Else $w_i=j$ and $j\neq k$, then draw $z_{j,i}$ according to $z_{j,i}\sim N(\mathbf{X}_\gamma\boldsymbol{\beta}_j,1)$ truncated right by the newly generated $z_{k,i}$, that is,

$$z_{j,i}\sim\mathcal{N}(\mathbf{X}_\gamma\boldsymbol{\beta}_j,1)1_{\{z_{j,i}\leq z_{k,i}\}}. \tag{13}$$

Endfor.

Here, we set $z_{K,i}\sim N(0,1)$ when $w_i=K$, that is, we introduce a new equation $z_{K,i}=\mathbf{X}_\gamma\boldsymbol{\beta}_K+e_{K,i}$, $i=1,\ldots,m$, with $\boldsymbol{\beta}_K$ being a zero vector and $e_{K,i}\sim N(0,1)$.

Algorithm 1

parameter $\boldsymbol{\beta}$ is still dependent on k and $\boldsymbol{\gamma}$, denoted by $\boldsymbol{\beta}_{k,\gamma}$. Then (5) is rewritten as

$$\mathbf{z}_k=\mathbf{X}_\gamma\boldsymbol{\beta}_{k,\gamma}+\mathbf{e}_k,\quad k=1,\ldots,K-1, \tag{6}$$

where $\mathbf{X}_\gamma$ means the column of $\mathbf{X}$ corresponding to those elements of $\boldsymbol{\gamma}$ that are equal to 1, and the same applies to $\boldsymbol{\beta}_{k,\gamma}$. Now, the problem is how to estimate $\boldsymbol{\gamma}$ and the corresponding $\boldsymbol{\beta}_{k,\gamma}$ and $\mathbf{z}_k$ for each equation in (6).

2.2. Bayesian variable selection

A Gibbs sampler is employed to estimate all the parameters. Given $\boldsymbol{\gamma}$ for equation k, the prior distribution of $\boldsymbol{\beta}_\gamma$ is $\boldsymbol{\beta}_\gamma\sim N(0,c(\mathbf{X}_\gamma^T\mathbf{X}_\gamma)^{-1})$ [22], where c is a constant (we set $c=10$ in this study). The detailed derivation of the posterior distributions of the parameters are given in [22]. Here, we summarize the procedure for Bayesian variable selection. Denote

$$S(\boldsymbol{\gamma},\mathbf{z}_k)=\mathbf{z}_k^T\mathbf{z}_k-\frac{c}{c+1}\mathbf{z}_k^T\mathbf{X}_\gamma(\mathbf{X}_\gamma{}^T\mathbf{X}_\gamma)^{-1}\mathbf{X}_\gamma{}^T\mathbf{z}_k, \tag{7}$$

where $k=1,\ldots,K-1$. Then the Gibbs sampling algorithm for estimating $\{\boldsymbol{\gamma},\boldsymbol{\beta}_k,\mathbf{z}_k\}$ is as follows. By straightforward computing, the posteriori distribution $p(\boldsymbol{\gamma}|\mathbf{z}_1,\ldots,\mathbf{z}_{K-1})$ is approximated by

$$\begin{aligned} &p(\boldsymbol{\gamma}|\mathbf{z}_1,\ldots,\mathbf{z}_{K-1}) \\ &\quad\propto p(\mathbf{z}_1,\ldots,\mathbf{z}_{K-1}|\boldsymbol{\gamma})\,p(\boldsymbol{\gamma}) \\ &\quad\propto (1+c)^{-(K-1)n_\gamma/2} \\ &\qquad\times\exp\left\{-\frac{1}{2}\sum_{k=1}^{K-1}S(\boldsymbol{\gamma},\mathbf{z}_k)\right\}\prod_{i=1}^{n}\pi_i^{\gamma_i}(1-\pi_i)^{1-\gamma_i}, \end{aligned} \tag{8}$$

and the posterior distribution $p(\boldsymbol{\beta}_{k,\gamma}|\mathbf{z}_k)$ is given by

$$\boldsymbol{\beta}_{k,\gamma}|\mathbf{z}_k,\mathbf{X}_\gamma\sim N(\mathbf{V}_\gamma\mathbf{X}_\gamma{}^T\mathbf{z}_k,\mathbf{V}_\gamma). \tag{9}$$

The Gibbs sampling algorithm for estimating $\boldsymbol{\gamma}$, $\{\boldsymbol{\beta}_{k,\gamma}\}$, and $\{\mathbf{z}_k\}$ is illustrated in Algorithm 1.

In this study, 12000 Gibbs iterations are implemented with the first 2000 as burn-in period. Then we obtain the Monte Carlo samples as $\boldsymbol{\gamma}^{(t)},\boldsymbol{\beta}_k^{(t)},\mathbf{z}_k^{(t)}$, $t=2001,\ldots,T$, where $T=10000$. Finally, we count the number of times that each gene appears in $\boldsymbol{\gamma}^{(t)}$, $t=2001,2002,\ldots,T$. The genes with the highest appearance frequencies play the strongest role in predicting the target gene. We will discuss some implementation issues of Algorithm 1 in Section 3.

2.3. Bayesian estimation using the strongest genes

Now, assume that the genes corresponding to nonzeros of $\boldsymbol{\gamma}$ are the strongest genes obtained by Algorithm 1. For fixed $\boldsymbol{\gamma}$, we again use a Gibbs sampler to estimate the probit regression coefficients $\boldsymbol{\beta}_k$ as follows: first, draw $\boldsymbol{\beta}_{k,\boldsymbol{\gamma}}$ according to (11), then draw $\mathbf{z}_k$ and iterate the two steps. In this study, 1500 iterations are implemented with the first 500 as the burn-in period. Thus, we obtain the Monte Carlo samples $\boldsymbol{\beta}_{k,\boldsymbol{\gamma}}^{(t)}, \mathbf{z}_k^{(t)}$, $t = 501, \ldots, \tilde{T}$. The probability of a given sample $\mathbf{x}$ under each class is given by

$$P(w = k|\mathbf{x}) = \frac{1}{\tilde{T}} \sum_{t=1}^{\tilde{T}} \prod_{j=1,\, j\neq k}^{K} \Phi\left(\mathbf{x}_{\boldsymbol{\gamma}} \boldsymbol{\beta}_{k,\boldsymbol{\gamma}}^{(t)} - \mathbf{x}_{\boldsymbol{\gamma}} \boldsymbol{\beta}_{j,\boldsymbol{\gamma}}^{(t)}\right), \quad k = 1, \ldots, K-1, \tag{14}$$

$$P(w = K|\mathbf{x}) = 1 - \sum_{k=1}^{K-1} P(w = k|\mathbf{x}), \tag{15}$$

where $\boldsymbol{\beta}_{K,\boldsymbol{\gamma}}^{(t)}$ is a zero vector; and the estimation of this sample is given by

$$\hat{w} \triangleq d(w) = \arg \max_{1 \le k \le K} P(w = k|\mathbf{x}). \tag{16}$$

Note that (15) may be computed using another formulation, which is replaced by [28, (13)].

In order to measure the fitting accuracy of such a predictor, we next define the coefficient of determination (COD) for this probit predictor. In fact, the above $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ (including all parameters $\boldsymbol{\beta}_{k,\boldsymbol{\gamma}}$) are dependent on the target gene w. Firstly, a probabilistic error measure $\epsilon(w, \mathbf{x}_{\boldsymbol{\gamma}}, \boldsymbol{\beta})$ associated with the predictors $\boldsymbol{\gamma}, \boldsymbol{\beta}$ is defined as

$$\epsilon(w, \mathbf{x}_{\boldsymbol{\gamma}}, \boldsymbol{\beta}) \triangleq \mathbb{E}\left[\, |\, d(w) - w\, |^2\right], \tag{17}$$

where $\mathbb{E}$ denotes the expectation. Similar to the definition in [14], the COD for w relative to the conditioning sets $\boldsymbol{\gamma}, \boldsymbol{\beta}$ is defined by

$$\theta = \frac{\epsilon - \epsilon(w, \mathbf{x}_{\boldsymbol{\gamma}}, \boldsymbol{\beta})}{\epsilon}, \tag{18}$$

where ϵ is the error of the best (constant) estimate of w in the absence of any conditional variables. In the case of minimum mean square error estimation, ϵ is defined as

$$\epsilon = \mathbb{E}[\, |\, w - g(\mathbb{E}(w))\, |^2], \tag{19}$$

where g is a $\{-1, 0, 1\}$-valued threshold function [$g(z) = 0$ if $-0.5 < z < 0.5$, $g(z) = 1$ if $z \ge 0.5$, and $g(z) = -1$ if $z \le -0.5$] for ternary data.

3. FAST IMPLEMENTATION ISSUES

The computational complexity of the Bayesian gene selection algorithm in (Algorithm 1) is very high. For example, if there are 1000 gene variables, then for each iteration, we have to compute the matrix inverse $(\mathbf{X}_{\boldsymbol{\gamma}}{}^T \mathbf{X}_{\boldsymbol{\gamma}})^{-1}$ 1000 times because we need to compute (10) for each gene. Hence, some fast algorithms must be developed to deal with the problem.

3.1. Preselection method

When there is a very large number of genes, we employ a preselection method. In pattern recognition, the following criterion is often adopted: the smaller is the sum of squares within groups and the bigger is the sum of squares between groups, the better is the classification accuracy. Therefore, we can define a score using the above two statistics to preselect genes, that is, the ratio of the between-group to within-group sum of squares. It is not necessary to adopt this procedure if the number of genes is small.

3.2. Computation of $p(\gamma_j|\mathbf{z}_k, \gamma_{i\neq j})$ in (10)

Because γ_j only takes 0 or 1, we can take a close look at $p(\gamma_j = 1|\mathbf{z}_k,\ i \neq j)$ and $p(\gamma_j = 0|\mathbf{z}_k,\ i \neq j)$. Let

$$\begin{aligned} \boldsymbol{\gamma}^1 &= (\gamma_1, \ldots, \gamma_{j-1},\ \gamma_j = 1,\ \gamma_{j+1}, \ldots, \gamma_n), \\ \boldsymbol{\gamma}^0 &= (\gamma_1, \ldots, \gamma_{j-1},\ \gamma_j = 0,\ \gamma_{j+1}, \ldots, \gamma_n). \end{aligned} \tag{20}$$

After a straightforward computation of (10), we have

$$p(\gamma_j = 1|\mathbf{z}_k,\ \gamma_{i\neq j}) \propto \frac{1}{1+h}, \tag{21}$$

with

$$h = \frac{1-\pi_j}{\pi_j} \exp\left\{\frac{S(\boldsymbol{\gamma}^1, \mathbf{z}_k) - S(\boldsymbol{\gamma}^0, \mathbf{z}_k)}{2}\right\} \sqrt{1+c}. \tag{22}$$

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

3.3. Fast computation of $S(\boldsymbol{\gamma}, \mathbf{z}_k)$ in (7)

From the above discussion, it is a key step to compute $S(\boldsymbol{\gamma}, \mathbf{z}_k)$ fast when a gene variable is added or removed from $\boldsymbol{\gamma}$. Denote

$$E(\boldsymbol{\gamma}, \mathbf{z}_k) = \mathbf{z}_k^T \mathbf{z}_k - \mathbf{z}_k^T \mathbf{X}_{\boldsymbol{\gamma}} (\mathbf{X}_{\boldsymbol{\gamma}}{}^T \mathbf{X}_{\boldsymbol{\gamma}})^{-1} \mathbf{X}_{\boldsymbol{\gamma}}{}^T \mathbf{z}_k, \tag{23}$$

where $k = 1, \ldots, K-1$. Then (23) can be computed using the fast QR-decomposition, QR-delete, and QR-insert algorithms when a variable is added or removed [29, Chapter 10.1.1b]. Now, we want to estimate $S(\boldsymbol{\gamma}, \mathbf{z}_k)$ in (7). Comparing (23) and (7), one can obtain the following equation:

$$\mathbf{z}_k^T \mathbf{X}_{\boldsymbol{\gamma}} (\mathbf{X}_{\boldsymbol{\gamma}}{}^T \mathbf{X}_{\boldsymbol{\gamma}})^{-1} \mathbf{X}_{\boldsymbol{\gamma}}{}^T \mathbf{z}_k = (1+c)[S(\boldsymbol{\gamma}, \mathbf{z}_k) - E(\boldsymbol{\gamma}, \mathbf{z}_k)]. \tag{24}$$

Substituting (24) into (7), after a straightforward computation, $S(\boldsymbol{\gamma}, \mathbf{z}_k)$ is given by

$$S(\boldsymbol{\gamma}, \mathbf{z}_k) = \frac{\mathbf{z}_k^T \mathbf{z}_k + cE(\boldsymbol{\gamma}, \mathbf{z}_k)}{1+c}, \quad k = 1, \ldots, K-1. \tag{25}$$

(i) Preselect genes.

(ii) Initialization: Randomly set initial parameters $\boldsymbol{\gamma}^{(0)}$, $\boldsymbol{\beta}_k^{(0)}$, $\mathbf{z}_k^{(0)}$.

(iii) For $t = 1, 2, \ldots, 12000$

Draw $\boldsymbol{\gamma}^{(t)}$. For $j = 1, \ldots, n$

Compute $S(\boldsymbol{\gamma}^{(t)}, \mathbf{z}_k)$ using QR-delete or QR-insert.

Compute $p(\gamma_j = 1 | \mathbf{z}_k, \gamma_{i \neq j}^{(t)})$ according to (21).

Draw $\gamma_j^{(t)}$ from $p(\gamma_j = 1 | \mathbf{z}_k^{(t-1)}, \gamma_{i \neq j}^{(t)})$.

Draw $\boldsymbol{\beta}_k^{(t)}$ according to (11);

Draw $\mathbf{z}_k^{(t)}$ according to (12) and (13).

(iv) Endfor.

(v) Count the frequency of each gene appeared in $\boldsymbol{\gamma}^{(t)}$, $t = 2001, \ldots, 12000$.

Algorithm 2

Thus, after computing $E(\boldsymbol{\gamma}, \mathbf{z}_k)$ using QR-decomposition, QR-delete, and QR-insert algorithms, we then obtain $S(\boldsymbol{\gamma}, \mathbf{z}_k)$. Here, we only need to compute the matrix inverse one time each iteration, but in the original algorithm, we have to compute the matrix inverse for n time each iteration. The computation complexity will be much smaller than that of the original algorithm [22] due to our processing techniques. To that end, we summarize our fast Bayesian gene selection algorithm as in Algorithm 2.

Notice that if it happens that the number of selected genes is more than the total number of samples, we need to remove this case because $(\mathbf{X}_{\boldsymbol{\gamma}}{}^T \mathbf{X}_{\boldsymbol{\gamma}})^{-1}$ does not exist. Another concern is that if it happens that $(\mathbf{X}_{\boldsymbol{\gamma}}{}^T \mathbf{X}_{\boldsymbol{\gamma}})$ is singular due to some rows or columns being a constant, then we need to add a very small random number to each element in $\mathbf{X}_{\boldsymbol{\gamma}}$.

4. EXPERIMENTAL RESULTS

In the first step in constructing a gene regulatory network, the complexity of the expression data is reduced by thresholding changes in transcript level into ternary expression data: -1 (down-regulated), $+1$ (up-regulated), or 0 (invariant). When using multiple microarrays, the absolute signal intensities vary extensively due to both the process of preparing and printing the EST elements [30] and the process of preparing and labeling the cDNA representations of the RNA pools. This problem is solved via internal standardization. We then build gene regulatory networks using the proposed approaches.

4.1. *Malignant melanoma microarray data*

The gene expression profiles used in this study result from a study of 31 malignant melanoma samples [24]. For the study, total messenger RNA was isolated directly from melanoma biopsies. Fluorescent cDNA from the message was prepared and hybridized to a microarray containing probes for 8 150 cDNAs (representing 6 971 unique genes). A set of 587 genes has been subjected to an analysis of their ability to cross predict each other's state in a multivariate setting [11, 13, 25]. From these, we have selected 26 differential genes using the following t-test:

$$t(j) = \frac{\bar{x}_{1,j} - \bar{x}_{2,j}}{s_0(j)\sqrt{1/m_1 + 1/m_2}}, \quad j = 1, \ldots, p, \tag{26}$$

with

$$s_0(j) \triangleq \sqrt{\frac{(m_1 - 1)s_1(j)^2 + (m_2 - 1)s_2(j)^2}{m_1 + m_2}}, \tag{27}$$

where p is the number of genes, $\{\bar{x}_{k,j}\}_{k=1}^2$ denotes the average expression level of gene j across the samples belonging to class k, m_1 and m_2 are the numbers of the two classes, and $\{s_k(j)^2\}_{k=1}^2$ are the variances of gene j across the samples belonging to class k. Genes with $t(j) \geq 0.05$ are listed in Table 1.

COD values for all the 26 targets have been computed using the strongest genes found via the Bayesian selection. CODs have been computed using leave-one-out cross validation. The strongest genes for each target are listed in the second column of Table 2 and the third column lists the CODs using the top 2, 3, and 4 genes for each target and using the probit regression to form the predictors. Several points should be noted. First, while the theoretical (distributional) COD values increase as the number of predictors increases, this is not necessarily the case for experimental data, especially when small samples are involved (on account of overfitting and high variance of cross-validation error estimation). Second, pirin (no. 2) is a strong predictor gene in many cases, and this agrees with the comment in the original paper that pirin has a very high discriminative weight [24]. Third, even with feature selection and a suboptimal predictor function, for the most part, the CODs are fairly high.

Having made the last point, we note that our salient interest is gene selection. Hence, having found strong genes via Bayesian variable selection, we are not compelled to use the probit regression model to form the predictors; rather, we can choose the optimal predictor using the strong genes among all possible (full-logic) predictor functions. We can

Table 1: The 26 differential genes.

Gene no.	Index no.	Gene description
1	3	Tumor protein D52
2	7	Pirin
3	14	V-myc avian myelocytomatosis viral oncogene homolog
4	42	Endothelin receptor type B
5	60	ESTS
6	79	Alpha-2-macroglobulin
7	117	V-myc avian myelocytomatosis viral oncogene homolog
8	126	ESTs
9	175	Myotubularin related protein 4
10	210	NGFI-A binding protein 2 (ERG1 binding protein 2)
11	216	IQ motif containing GTPase activating protein 1
12	220	Annexin A2
13	228	ESTs
14	245	Homo sapiens mRNA; cDNA DKFZp434L057 (from clone DKFZp434L057)
15	282	Endothelin receptor type B
16	292	ESTs
17	323	ESTs
18	360	Glycoprotein M6B
19	372	"Nuclear receptor subfamily 4, group A, member 3"
20	374	Thrombospondin 2
21	387	"ESTs, weakly similar to HP1-BP74 protein [M.musculus]"
22	404	"Phosphofructokinase, liver"
23	506	Placental transmembrane protein
24	556	Human insulin-like growth factor binding protein 5 (IGFBP5) mRNA
25	573	"Platelet-derived growth factor receptor, alpha polypeptide"
26	576	ESTs

Table 2: Strongest genes to predict each gene and the corresponding COD values for 2, 3, and 4 predictor genes.

Target gene no.	Strongest genes (no.)				COD		
	1	2	3	4	2	3	4
1	19	23	22	17	0.6452	0.6129	0.7097
2	25	1	19	11	0.3871	0.6774	0.8065
3	7	23	2	5	0.7097	0.7742	0.7742
4	15	2	13	17	0.7419	0.7742	0.8710
5	14	2	13	10	0.5484	0.5161	0.4194
6	10	2	19	24	0.6129	0.7097	0.8387
7	3	2	17	1	0.7419	0.8387	0.8387
8	20	2	21	14	0.5161	0.5484	0.5484
9	2	13	17	15	0.6774	0.7097	0.7742
10	6	20	2	4	0.6129	0.6452	0.6774
11	13	25	2	1	0.8710	0.8710	0.7742
12	2	13	11	14	0.6452	0.6452	0.7419
13	2	15	11	18	0.8387	1.0000	1.0000
14	2	25	21	15	0.6774	0.7742	0.6774
15	2	4	13	14	0.8065	0.7419	0.9677
16	4	25	2	7	0.6452	0.7097	0.6452
17	11	18	2	8	0.8387	0.8065	0.8387
18	2	17	13	23	0.8387	0.7742	0.8710
19	1	22	2	9	0.7419	0.6774	0.7419
20	22	5	10	24	0.3548	0.3548	0.7419
21	25	2	14	20	0.7742	0.7742	0.7742
22	2	9	6	23	0.6774	0.7097	0.7742
23	24	2	1	5	0.5161	0.5484	0.6774
24	2	20	3	7	0.5806	0.6129	0.6452
25	11	2	14	13	0.7742	0.6774	0.8065
26	17	13	2	23	0.7742	0.7742	0.8387

Table 3: Three-predictor COD values using full-logic predictor, full search, and Bayesian-selected genes. There are 2300 three-predictor sets for each target gene.

Target gene no.	Probit position	logic COD (best)	logic COD (probit)
1	32	0.8065	0.7419
2	59	0.8387	0.7419
3	36	0.9355	0.9032
4	15	0.9677	0.9032
5	52	0.7742	0.6774
6	1	0.9677	0.9677
7	30	0.9355	0.9032
8	91	0.8387	0.7419
9	141	0.8710	0.7742
10	25	0.9677	0.9032
11	49	0.9677	0.8710
12	173	0.8387	0.7419
13	1	1.0000	1.0000
14	212	0.8387	0.7419
15	102	0.9677	0.9355
16	46	0.8710	0.7742
17	12	0.9677	0.9355
18	289	0.9355	0.8710
19	196	0.9677	0.8387
20	21	0.8710	0.8387
21	14	0.8387	0.8065
22	16	0.9355	0.9032
23	48	0.9032	0.8065
24	29	0.8065	0.7097
25	69	0.8710	0.7742
26	49	0.9355	0.9032

also compare the COD for this approach with the fully optimal COD derived from considering all possible predictor sets from among the full-gene set and all possible predictor functions. The results of this analysis for three predictor variables are shown in Table 3. For each target, the second column gives the rank of the COD resulting from the probit predictors in the list of all the 2300 CODs found from all possible subsets of three predictors using the best full-logic predictor. The selected gene sets rank very high except in a couple of cases. The third and fourth columns give the CODs for the best full-logic predictor with a full search of the gene subsets and the best full-logic predictor using the strongest three genes found by Bayesian gene selection. As must be the case, the values in the third column must exceed the values in the fourth, but in general, this does not happen much, even when the probit-selected predictor set does not rank near the top. The differences are likely due to multivariate interaction between the predictors not recognized by the sequential selection of strongest genes [17]. Table 4 shows analogous results for four predictors. For it, we note that there are 12 650 predictor sets for each target. Similar comments apply to the genes in Table 4.

It is interesting to compare the fourth column in Table 4 with the third in Table 3. For large gene sets (say, 600 to 1000 genes), a full search over all the three-variable predictor sets is feasible with a supercomputer running for weeks [15]. But a full search is not feasible for a full search over all four-variable predictor sets. Optimal four-connectivity may not be possible in network design. Hence, the small loss in COD between the full-search column in Table 3 and the probit-selection column in Table 4 demonstrates the potential of the Bayesian feature selection. Indeed, there are a number of cases in which the four-variable probit-selected genes outperform the corresponding three-variable full-search genes. Just to get an idea of the vast difference between the methods, the Gibbs sampler would need approximately 12000×1000 iterations, whereas the fully optimal full-search predictor would need to consider 2^{1000} predictor sets. Even for four-variable predictor sets, the full search needs C_4^{1000} iterations, which is vastly larger than the Gibbs sampling search.

Table 4: Four-Predictor COD values using full-logic predictor, full search, and Bayesian-selected genes. There are 12650 four-predictor sets for each target gene.

Target gene no.	Probit position	Logic COD (best)	Logic COD (probit)
1	48	0.8710	0.7742
2	70	0.8710	0.8065
3	14	0.9677	0.9355
4	283	1.0000	0.9355
5	48	0.8387	0.7419
6	1	0.9677	0.9677
7	82	0.9677	0.9032
8	101	0.8710	0.7742
9	60	0.9032	0.8387
10	569	0.9677	0.8710
11	82	0.9677	0.9032
12	510	0.9355	0.8065
13	1	1.0000	1.0000
14	131	0.8710	0.8065
15	1	1.0000	1.0000
16	60	0.8710	0.8065
17	65	0.9355	0.8710
18	364	0.9677	0.8710
19	170	0.8065	0.7419
20	52	0.9355	0.8387
21	193	0.9355	0.9032
22	163	0.9677	0.9032
23	240	0.9677	0.8710
24	91	0.8065	0.7419
25	58	0.9032	0.8387
26	79	0.9677	0.9355

5. CONCLUSION

We have studied the problem of multilevel gene prediction and genetic network construction from gene expression data based on multinomial probit regression with Bayesian gene selection, which selects genes closely related to a particular target gene. Some fast implementation issues for this Bayesian gene selection method have been discussed, in particular, computing estimation errors recursively using QR decomposition. Experimental results using malignant melanoma data show that the Bayesian gene selection yields predictor sets with coefficients of determination that are competitive with those obtained via a full search over all possible predictor sets.

ACKNOWLEDGMENTS

This research was supported by the National Human Genome Research Institute and the Translational Genomics Research Institute. X. Wang was supported in part by the US National Science Foundation under Grant DMS-0225692.

REFERENCES

[1] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Baysian networks to analyze expression data," *Computational Biology*, vol. 7, no. 3/4, pp. 601–620, 2000.

[2] E. J. Moler, D. C. Radisky, and I. S. Mian, "Integrating naive Bayes models and external knowledge to examine copper and iron homeostasis in S. cerevisiae," *Physiological Genomics*, vol. 4, no. 2, pp. 127–135, 2000.

[3] K. Murphy and S. Mian, "Modelling gene expression data using dynamic Bayesian networks," Tech. Rep., University of California, Berkeley, Calif, USA, 1999, http://citeseer.nj.nec.com/murphy99modelling.html.

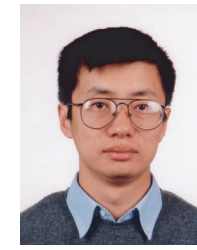
[4] D. Pe'er, A. Regev, G. Elidan, and N. Friedman, "Inferring subnetworks from perturbed expression profiles," *Bioinformatics*, vol. 17, suppl. 1, pp. S215–S224, 2001.

[5] T. Akutsu, S. Miyano, and S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under Boolean network model," in *Proc. Pacific Symposium on Biocomputing*, vol. 4, pp. 17–28, Maui, Hawaii, USA, January 1999.

[6] P. D'haeseleer, S. Liang, and R. Somogyi, "Genetic network inference: from co-expression clustering to reverse engineering," *Bioinformatics*, vol. 16, no. 8, pp. 707–726, 2000.

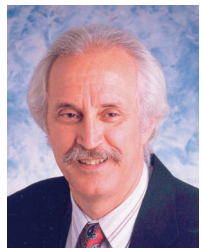
[7] S. Huang, "Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorgenesis and drug discovery," *Molecular Medicine*, vol. 77, no. 6, pp. 469–480, 1999.
[8] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, NY, USA, 1993.
[9] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.
[10] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Gene perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, no. 10, pp. 1319–1331, 2002.
[11] S. Kim, H. Li, E. R. Dougherty, et al., "Can Markov chain models mimic biological regulation?," *Biological Systems*, vol. 10, no. 4, pp. 337–357, 2002.
[12] X. Zhou, X. Wang, and E. R. Dougherty, "Construction of genomic networks using mutual-information clustering and reversible-jump Markov-Chain-Monte-Carlo predictor design," *Signal Processing*, vol. 83, no. 4, pp. 745–761, 2003.
[13] S. Kim, E. R. Dougherty, Y. Chen, et al., "Multivariate measurement of gene expression relationships," *Genomics*, vol. 67, no. 2, pp. 201–209, 2000.
[14] E. R. Dougherty, S. Kim, and Y. Chen, "Coefficient of determination in nonlinear signal processing," *Signal Processing*, vol. 80, no. 10, pp. 2219–2235, 2000.
[15] E. B. Suh, E. R. Dougherty, S. Kim, D. E. Russ, and R. L. Martino, "Parallel computing methods for analyzing gene expression relationships," in *Proc. SPIE Microarrays: Optical Technologies and Informatics*, San Jose, Calif, USA, January 2001.
[16] I. Tabus and J. Astola, "On the use of MDL principle in gene expression prediction," *Applied Signal Processing*, vol. 2001, no. 4, pp. 297–303, 2001.
[17] R. F. Hashimoto, E. R. Dougherty, M. Brun, Z.-Z. Zhou, M. L. Bittner, and J. M. Trent, "Efficient selection of feature sets possessing high coefficients of determination based on incremental determinations," *Signal Processing*, vol. 83, no. 4, pp. 695–712, 2003.
[18] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, 2002.
[19] R. Jornsten and B. Yu, "Simultaneous gene clustering and subset selection for sample classification via MDL," *Bioinformatics*, vol. 19, no. 9, pp. 1100–1109, 2003.
[20] T. R. Golub, D. K. Slonim, P. Tamayo, et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.
[21] H. Chipman, E. I. George, and R. McCulloch, "The practical implementation of Bayesian model selection," in *Model Selection*, vol. 38, pp. 65–134, Institute of Mathematical Statistics, Hayward, Calif, USA, 2001.
[22] K. E. Lee, N. Sha, E. R. Dougherty, M. Vannucci, and B. K. Mallick, "Gene selection: a Bayesian variable selection approach," *Bioinformatics*, vol. 19, no. 1, pp. 90–97, 2003.
[23] J. Albert and S. Chib, "Bayesian analysis of binary and polychotomous response data," *Journal of the American Statistical Association*, vol. 88, no. 422, pp. 669–679, 1993.
[24] M. Bittner, P. Meltzer, Y. Chen, et al., "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, pp. 536–540, 2000.
[25] S. Kim, E. R. Dougherty, M. L. Bittner, et al., "General nonlinear framework for the analysis of gene interaction via multivariate expression arrays," *Biomedical Optics*, vol. 5, no. 4, pp. 411–424, 2000.
[26] K. Imai and D. A. van Dyk, "A Bayesian analysis of the multinomial probit model using marginal data augmentation," http://www.princeton.edu/~kimai/research/mnp.html.
[27] C. P. Robert, "Simulation of truncated normal variables," *Statistics and Computing*, vol. 5, pp. 121–125, 1995.
[28] P. Yau, R. Kohn, and S. Wood, "Bayesian variable selection and model averaging in high-dimensional multinomial nonparametric regression," *Computational and Graphical Statistics*, vol. 12, no. 1, pp. 23–54, 2003.
[29] G. A. F. Seber, *Multivariate Observations*, John Wiley & Sons, NY, USA, 1984.
[30] Y. Chen, E. R. Dougherty, and M. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *Journal of Biomedical Optics*, vol. 2, no. 4, pp. 364–374, 1997.

Xiaobo Zhou received the B.S. degree in mathematics from Lanzhou University, Lanzhou, China, in 1988, the M.S. and the Ph.D. degrees in mathematics from Peking University, Beijing, China, in 1995 and 1998, respectively. From 1988 to 1992, he was a Lecturer at the Training Center in the 18th Building Company, Chongqing, China. From 1992 to 1998, he was a Research Assistant and Teaching Assistant in the Department of Mathematics at Peking University, Beijing, China. From 1998 to 1999, he was a postdoctoral fellow in the Department of Automation at Tsinghua University, Beijing, China. From January 1999 to February 2000, he was a Senior Technical Manager of the 3G Wireless Communication Department at Huawei Technologies Co., Ltd., Beijing. From February 2000 to December 2000, he was a postdoctoral fellow in the Department of Computer Science at the University of Missouri-Columbia, Columbia, Mo. From January 2001 to September 2003, he was a postdoctoral fellow in the Department of Electrical Engineering at Texas A&M University, College Station, Tex. Since October 2003, he has been a postdoctoral fellow in the Harvard Center for Neurodegeneration and Repair in Harvard University Medical School and Radiology Department in Brigham and Women's Hospital. His current research interests include bioinformatics in genetics, protein structure informatics, imaging genetics, and gene transcriptional regulatory networks.



Xiaodong Wang received the B.S. degree in electrical engineering and applied mathematics (with the highest honor) from Shanghai Jiao Tong University, Shanghai, China, in 1992; the M.S. degree in electrical and computer engineering from Purdue University in 1995; and the Ph.D. degree in electrical engineering from Princeton University in 1998. From July 1998 to December 2001, he was an Assistant Professor in the Department of Electrical Engineering, Texas A&M University. In January 2002, he joined the Department of Electrical Engineering, Columbia University, as an Assistant Professor. Dr. Wang's research interests fall in the general areas of computing, signal processing, and communications. He has worked in the areas of digital communications, digital signal processing, parallel and distributed

computing, nanoelectronics, and bioinformatics, and has published extensively in these areas. His current research interests include wireless communications, Monte Carlo based statistical signal processing, and genomic signal processing. Dr. Wang received the 1999 NSF CAREER Award and the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award. He currently serves as an Associate Editor for the IEEE Transactions on Communications, the IEEE Transactions on Wireless Communications, the IEEE Transactions on Signal Processing, and the IEEE Transactions on Information Theory.



Edward R. Dougherty is a Professor in the Department of Electrical Engineering at Texas A&M University in College Station. He holds an M.S. degree in computer science from Stevens Institute of Technology in 1986 and a Ph.D. degree in mathematics from Rutgers University in 1974. He is the author of eleven books and the editor of other four books. He has published more than one hundred journal papers, is an SPIE Fellow, and has served as an Editor of the Journal of Electronic Imaging for six years. He is currently Chair of the SIAM Activity Group on Imaging Science. Prof. Dougherty has contributed extensively to the statistical design of nonlinear operators for image processing and the consequent application of pattern recognition theory to nonlinear image processing. His current research focuses on genomic signal processing, with the central goal being to model genomic regulatory mechanisms. He is Head of the Genomic Signal Processing Laboratory at Texas A&M University.