A MM-based Optimization Algorithm for Sparse Linear Modeling on Microarray Data Analysis

Xiaohui Chen
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, BC V6T 1Z4
Email: xiaohuic@ece.ubc.ca

Raphael Gottardo
Department of Statistics
University of British Columbia
Vancouver, BC V6T 1Z2
Email: raph@stat.ubc.ca

Abstract—Sparsity is crucial for high-dimensional statistical modeling. On one hand, dimensionality reduction can reduce the variability of estimation and thus provide reliable predictive power. On the other hand, the selected sub-model can discover and emphasize the underlying dependencies, which is useful for objective interpretation. Many variable selection methods have been proposed in literatures. For a prominent example, Least Absolute Shrinkage and Selection Operator (lasso) in linear regression context has been extensively explored. This paper discusses a class of scaled mixture of Gaussian models from both a penalized likelihood and a Bayesian regression point of view. We propose an Majorize-Minimize (MM) algorithm to find the Maximum A Posteriori (MAP) estimator, where the EM algorithm can be stuck at local optimum for some members in this class. Simulation studies show the outperformance of proposed algorithm in nonstochastic design variable selection scenario. The proposed algorithm is applied to a real large-scale E.coli data set with known bona fide interactions for constructing sparse gene regulatory networks. We show that our regression networks with a properly chosen prior can perform comparably to state-of-the-art regulatory network construction algorithms.

I. INTRODUCTION

Gene regulatory networks have been intensively studied over the last decade because of their inherent scientific and medical importance (e.g., in the fight against cancer) as well as the development of high-throughput measurement devices. The typical formats of data generated by these high-throughput experiments are gene microarray expression data. Specially, each microarray is used to capture the snapshot of the transcriptional status of cells, and one experiment usually includes several microarrays to reflect the change of status of a biological system over times. A typical situation of performing a microarray experiment is that we profile hundreds of thousands of genes on one microarray and have only a few hundreds of microarrays in one experiment. Moreover, recently studies [1] showed that gene regulatory networks have some global architectures. For example, many genes have very low connectivity with others (sparsity) and only a few hub genes control many downstream genes (hierarchy). Therefore, detecting the bona fide interactions among thousands of genes from a few hundreds of measurements is a challenging statistical task.

Throughout this paper, the assumed model for regulatory network is a multivariate linear regression network:

\[ y = X\beta + e \]  

(1)

where

- \( X_{n \times p} \) is the design matrix consisting of \( n \) measured expression values for \( p \) candidate regulators
- \( y_{n \times 1} \) is the response vector of target gene expression values
- \( \beta_{p \times 1} \) is the regulatory strength: \( \beta > 0 \) means up-regulation, \( \beta < 0 \) means down-regulation, and \( \beta = 0 \) represents no regulation
- \( e_{n \times 1} \) is an error term assumed to be Gaussian with constant variance \( \sigma^2 \), i.e. \( N(0, \sigma^2I) \).

To capture the dependency between the regulators (predictors) and target gene (response), we consecutively regress the expression levels of every target gene to \( X \), while keeping \( X \) fixed for all. For high-dimensional microarray data \( (p >> n) \), the assumed linear model may be redundant in the sense that not all of its predictors have significant effects on the response. Typically, there are hundreds or even thousands candidate regulators for one gene, which may include Transcription Factors (TFs) and microRNAs. Thus the full linear model is usually over-parameterized. The prediction error would decrease as including more coefficients and the fit would improve. However, the improvement is very marginal compared with the cost of estimating a much more complex model. Furthermore, since the Least Squares (LS) procedure rarely yields zero estimates for coefficients, the interpretation for the full model becomes unclear for separating the true predictors from the irrelevant variables. Hence, selecting a proper subset of predictors from the full model is crucial for high-dimensional statistical modeling. To achieve this variable selection purpose, it is generally to penalize more complex models and prefer simpler models. Here in the context of linear regression, our objective function is a penalized least squares and we want to minimize this objective function over \( \beta \) and \( \lambda_j \)

\[ O(\beta, \lambda|X, y) = \frac{1}{2}||y - X\beta||^2_2 + \sum_{j=1}^{p} \lambda_j (||\beta_j||), \]  

(2)

where \( p\lambda_j (||\beta_j||) \) is the penalty term for each large \( \beta \) absolute value, \( \lambda_j \) is the corresponding shrinkage amount, and \( || \cdot ||_2 \) is the Euclidean norm of a vector. Generally speaking, large
penalties tend to shrink the $\beta$ coefficients to zeros and without loss of generality, we assume $p_{x_j}$ takes the same functional form for all $j = 1 : p$, i.e. $p_{x_j} = p_{\lambda}$ in Eq.(2). For instance, choosing $p_{x_j} = \lambda |\beta_j|^p$ yields the $L_p$ penalty and the resulting model is also called bridge regression [2], [3]. A special case when $p = 1$ is the famous lasso model [4].

The contribution of this paper is two-fold: first, we propose a Majorize-Minimize (MM) algorithm to solve the minimization problem Eq.(2) for penalties/priors in the scaled mixture of Gaussian family. Secondly, we extended the proposed MM algorithm for the Normal-Jeffreys and Normal-Gamma priors in this scaled mixture of Gaussian family to tackle the ill-defined density at 0.

II. Method

A. A Bayes Model Perspective

Alternatively for Eq.(2), the penalized likelihood function has Bayes interpretation. The penalty term can be viewed as the prior regularization of the sampling model. Therefore, to maximize the penalized likelihood function is equivalent to find the mode of corresponding posterior distribution. For example, the $L_1$ estimator is the posterior mode resulted from Gaussian likelihood coupled with double-exponential (a.k.a. Laplacian) prior. Griffin and Brown [5], [6] proposed a family of scaled mixture of Gaussian priors on regression coefficients. In other words, the marginal prior distribution of regression coefficients in this family can be factored as a scaled mixture of Gaussian density:

$$
\pi(\beta_j) = \int \mathcal{N}(\beta_j; 0, \tau_j^2) p(\tau_j^2) d\tau_j^2,
$$

(3)

where $\mathcal{N}(x; \mu, \sigma^2)$ denotes the normal distribution of $x$, with mean $\mu$ and variance $\sigma^2$. Depending on different prior distribution specified for $\tau_j^2$, we can introduce different prior models. In this paper, we consider the Normal-Jeffreys (NJ), Normal-Gamma (NG), and Normal-Inverse-Gaussian (NIG) priors, which are corresponding to choose $p(\tau_j^2)$ as Jeffreys, Gamma, and Inverse-Gaussian distributions, respectively. The penalties are summarized in Table I. For details for these models, see [7], [5], [6]. In addition to the priors in this scaled mixture of Gaussian family, we also include the Smoothly Clipped Absolute Deviation (SCAD) penalty [8] in this paper, which is not on the menu of this class. However, it has nice property to avoid the over-penalization of $\beta$ with large absolute values which is the case in $L_1$.

B. Optimization algorithm

After defining models with different priors, the subsequent problem becomes finding $\hat{\beta}$ such that

$$
(\hat{\beta}, \hat{\lambda}) = \arg \min_{\beta, \lambda} \mathcal{O}(\beta, \lambda; X, y).
$$

(4)

$\lambda$ can be determined on a grid by cross-validation (CV) giving the smallest prediction errors.

It turns out the minimization of the objective function, $\mathcal{O}(\cdot)$, is not trivial. For $p_{\lambda}(\cdot) = \lambda |\cdot|$ (a.k.a. the lasso [4], it can be efficiently solved by Osborne’s [9] and the Least Angle Regression (LARS) algorithm [10]. However, these algorithms are not generalized to penalties other than $L_1$ . [5], [7] and [11] proposed an Expectation-Maximization (EM) algorithm to find the Maximum A Posteriori (MAP) estimator for general penalty functions. However, the EM algorithm is a sub-optimum algorithm for this scaled mixture of Gaussian family in the sense that the coefficients have to be excluded from subsequent iterations once they were estimated to be zeros. Motivated by this difficulty, Hunter and Li [12] proposed a Majorize-Minimize (MM) algorithm to optimize an approximated objective function for $L_1$ and SCAD penalties and successfully avoid the local optimum problem with the EM algorithm. In this paper, we extended the idea of their MM algorithm (we call it MM2 algorithm since the EM is a special version of MM algorithm which we call it MM1 for simplicity) and propose a new MM algorithm (MM3) to handle the general penalizations with infinity penalty (c.f. Table I).

Briefly speaking, the MM algorithm is a generalization of the EM algorithm [13]. It is an iterative two-stage optimization algorithm like the EM. The primary difference from the EM algorithm is that MM constructs a flexible surrogate function $\Phi$ which transfers the optimization problem of the original objective function $p_{\lambda}$. For our MM3 algorithm, we propose a surrogate function

$$
\Phi(\theta, \theta_0) = p_{\lambda}(\theta) + (\theta^2 - \theta_0^2) / 2(\theta_0 + \epsilon).
$$

(5)

for our perturbed penalty function $p_{\lambda}(\cdot)$ evaluated at $x$ and $p'_{\lambda}(\cdot)$ denotes the derivative of penalty function $p_{\lambda}(\cdot)$ as $x \rightarrow \theta$ from above.

Here, adding a small $\epsilon > 0$ perturbation to the original penalty $p_{\lambda}$ avoids the ill-defined (infinity) density at 0 for NJ and NG priors. Due to space limitation, we refer the readers to the details in [14].

III. A SIMULATION STUDY

To compare the performances of models with different penalizations and algorithmic solutions, we simulate 100 data

| TABLE I PENALIZATIONS/LOG(PRIOR) AND THEIR FIRST ORDER DERIVATIVES EVALUATED AT $|\beta_j|$. NJ AND NG PRIORS HAVE INFINITY DENSITY AT 0. | $p_{\lambda}(\beta_j)$ | $\lambda |\beta_j|$ | $\log |\beta_j|$ | $\log q_j - \log K_1(\epsilon q_j)$ |
| --- | --- | --- | --- | --- |
| LASSO | $\lambda |\beta_j|$ | $\lambda |\beta_j|$ | $\log |\beta_j|$ | $\log q_j - \log K_1(\epsilon q_j)$ |
| NJ | $\frac{1}{2} \lambda (|\beta_j| - \log \sqrt{2\pi |\beta_j|} + \frac{\lambda^2}{2})$ | $\frac{1}{2} \lambda (|\beta_j| - \log \sqrt{2\pi |\beta_j|} + \frac{\lambda^2}{2})$ | $\log q_j - \log K_1(\epsilon q_j)$ |
| SCAD | $\lambda |\beta_j|/(|\beta_j| + 2\lambda^2) - \lambda |\beta_j|/(\alpha |\beta_j| + 2\lambda^2)$ | $\lambda |\beta_j|/(|\beta_j| + 2\lambda^2) - \lambda |\beta_j|/(\alpha |\beta_j| + 2\lambda^2)$ | $\log q_j - \log K_1(\epsilon q_j)$ |

Note: $q_j = \sqrt{(\lambda^2/\pi)^2 + (\lambda^2/\pi)^2}$. $K_1$ is the modified Bessel function of the second kind.
sets, each with \( n = 100 \) points and \( \sigma = 1 \). Following the setup of [12], the true coefficients are set to be \( \beta = (3, 0, 0, 0, 1.5, 0, 0, 0, 2, 0, 0, 0, 0, 0, 1.5, 0, 0, 0, 0, 0, 0, 0)^T \), with \( p = 12 \). We introduce three correlation levels between covariates: \( \rho = 0.1, 0.5, 0.9 \). Data \( X \) is scaled to mean 0 and unit variance and \( y \) is centered. All models and algorithms are initialized with all 0’s (an extreme situation). Optimal parameters are determined by 5-fold CV as used in [4]. The accuracy of various models are measured by the number of correctly estimated non-zeros (significant variables), which is equal to 3. Note that we do not implement the SCAD prior with the EM algorithms. The simulation result is shown in Fig. 1. It is easy to observe that the MM3 uniformly dominates the EM and the MM2 in terms of the number of correctly identified non-zeros. Comparing across models, the EM is always stuck at 0’s for all models and the MM2 only selects variables in case of \( L_1 \) and SCAD. For MM3, starting from zeros has no significant effect on the selected variables. Hence, the local optimum issue can be well handled by the MM3 in this extreme situation.

IV. Results on a E.coli Data Set

In [15], Faith et.al. showed how RegulonDB database [16] can serve as a ground truth of regulatory network for E. coli. RegulonDB contains 3216 experimentally confirmed regulatory interactions among 1058 genes and 153 TFs. [15] assembled a compendium of 445 new and previously published E. coli K12 Affymetrix Antisense2 microarray expression profiles collected under various conditions. Compared with the ground truth (see Fig. 2), this compendium is an ideal real dataset that we can evaluate the performance of various models.

We also compare our linear models with the state-of-the-art gene regulatory network construction algorithms, i.e. Context Likelihood Relatedness (CLR) algorithm [15]. CLR is a mutual information (MI)-based method. The CLR algorithm estimates the likelihood of the MI score for a particular pair of genes, \( i \) and \( j \), by comparing the MI value for that pair of genes to a background distribution of MI values (the null model). The performances of all methods are measured by the area under the precision-recall curve (PR curve). Precision is defined as the fraction of true positives (TP) out of the total predicted positives, and recall the fraction of true positives out of the actual total positives. By formulation, they can be expressed as following:

\[
\text{precision} = \frac{TP}{TP + FP}, \quad (7) \\
\text{recall} = \frac{TP}{TP + FN}. \quad (8)
\]

Hence, the larger area under the PR curve, the better performance an algorithm has.

From the PR curve in Figure 3, we can see that the \( L_1 \) penalty fitted with the EM and the MM3, the SCAD penalty with the MM3, and the NIG penalty with the EM, perform best in most algorithms and are comparable with the CLR algorithm. The NJ prior for all three algorithms are essentially overlapping each other. Performances of the NG prior for three algorithms are very close, although they are all inferior to the CLR algorithm. Finally, we retrieve the estimated network from \( L_1 \) penalty with 60% precision, and the network is shown in Fig. 4. Clearly, the learned network is sparse and there are a few hub genes controls other genes. Our learned newtork agrees well with the general characteristics of gene regulatory networks.

V. Conclusion

In this paper, we proposed a new MM algorithm for solving the penalized regression with scaled mixture of Gaussian pri-
Fig. 3. Precision-recall curves for various models and algorithms on *E.coli* data set. MI is the raw MI method. Z is CLR algorithm.

(a) $L_1$  (b) SCAD  
(c) NJ  (d) NG  
(e) NIG

ors and SCAD penalty. By weakening the regularity conditions on the penalty function, our algorithm can handle extreme situation with priors that have unbounded densities at origin. It allows the variables to be re-introduced after it has been removed from model. The new algorithms can be applied to a wide range of penalty functions, which do not have to retain the finiteness and integrability assumption. Since our MM3 algorithms allows the infinity penalty at zero, we suggest the proposed algorithm has the ability to identify very sparse signals from potentially huge dimensions. Simulation studies demonstrated that the proposed algorithm can handle ill-defined initialization. A real biological data set with *bona fide* interactions known is used to evaluate linear regression models. The PR curves showed that $L_1$ MM3, SCAD MM3, $L_1$ EM, and NIG EM methods are performing best among all models, and their performances are comparable to the state-of-the-art regulatory network construction algorithm CLR.

ACKNOWLEDGMENT

This work was supported in part by the Canadian Institutes of Health Research (CIHR).

REFERENCES


