11-25-2010 12:00 AM

Cost-efficient Variable Selection Using Branching LARS

Li Hua Yue
The University of Western Ontario

Supervisor
Dr. Duncan Murdoch
The University of Western Ontario Joint Supervisor
Dr. Wenqing He
The University of Western Ontario

Graduate Program in Statistics and Actuarial Sciences
A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy

Recommended Citation

Yue, Li Hua, "Cost-efficient Variable Selection Using Branching LARS" (2010). *Electronic Thesis and Dissertation Repository*. 43.
https://ir.lib.uwo.ca/etd/43

Cost-efficient Variable Selection Using Branching LARS

(Spine title: Cost-efficient Variable Selection Using Branching LARS)
(Thesis format: Monograph)

by

Li Hua Yue

Collaborative Biostatistics Program
in
Department of Statistical & Actuarial Sciences
Department of Epidemiology and Biostatistics

Submitted in partial fulfillment
of the requirements for the degree of Doctor of Philosophy

School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

THE UNIVERSITY OF WESTERN ONTARIO

SCHOOL OF GRADUATE AND POSTDOCTORAL STUDIES

CERTIFICATE OF EXAMINATION

Supervisors

Dr. Duncan Murdoch

Dr. Wenqing He

Examiners

Dr. John Braun

Dr. Yun-Hee Choi

Dr. Rafal Kustra

Dr. Adam Metzler

The thesis by

Li Hua Yue

entitled:

Cost-efficient Variable Selection Using Branching LARS

is accepted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

Date

Chair of the Thesis Examination Board

ABSTRACT

Variable selection is a difficult problem in statistical model building. Identification of cost efficient diagnostic factors is very important to health researchers, but most variable selection methods do not take into account the cost of collecting data for the predictors. The trade off between statistical significance and cost of collecting data for the statistical model is our focus. A Branching LARS (BLARS) procedure has been developed that can select and estimate the important predictors to build a model not only good at prediction but also cost efficient. BLARS method is an extension of the LARS variable selection method to incorporate various costs of factors, where branch and bound search method is employed to accelerate the search process. Both additive and non-additive costs will be addressed. The R package *branchLars* which implements BLARS will be described. We will show that a “cheaper” model could be selected by sacrificing a user selected amount of model accuracy.

Keywords: Variable selection, Cost efficient, BLARS, LARS, Lasso, Branch and bound

ACKNOWLEDGEMENT

Lots of thanks go to my supervisors Dr. Duncan Murdoch and Dr. Wenqing He, who gave me the opportunity to perform researches with them. Being able to learn from them is one of the most valuable experiences in my life. They have given great help on the thesis topic, research setup, application development, data analysis, and software building. It is impossible to have finished this thesis without their help.

In addition, I would like to thank Dr. Hristo Sendov. Taking his suggestion has improved the time efficiency and effectiveness of the BLARS search procedure.

Special thanks to Dr. Evelyn Vingilis and Ms. Jane Seeley from the Department of Family Medicine at Schulich School of Medicine & Dentistry. They provided me generous support in conducting data analysis of the Assertive Community Treatment(ACT) project which was composed to one chapter of my thesis.

Furthermore, I would like to thank the examiners, Dr. John Braun, Dr. Rafal Kustra, Dr. Yun-Hee Choi and Dr. Adam Metzler, for reviewing my thesis and making suggestions.

Many thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC). Supported by postgraduate scholarships (PGS D and CGS M) from the NSERC, I could concentrate on the research and make progress matching the planned schedule.

The rest of the thanks go to my family. The spiritual and financial support that they have given me has always been tremendous.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Introduction

Variable selection is a difficult problem in statistical model building, especially when a large number of variables are under consideration. If many insignificant variables are selected, the statistical model will lose its predictive power and the results will be hard to interpret. In practice, a desirable variable selection procedure should select variables consistently and result in a parsimonious model (simpler models are preferred). With large data sets, computation time for variable selection should also be taken into account. Computational efficiency is another indicator for a good variable selection approach. These are the guidelines when we develop a new variable selection method.

We have benefited from great improvements in variable selection in recent years. Many new approaches, such as lasso (Tibshirani, 1996) and LARS (Efron *et al.*, 2004), can efficiently select variables and estimate significant variable effects, and result in more accurate parsimonious models. We integrate these new methods into our variable selection strategy for their merits, but we add in a cost element which is rarely considered in the literature. It is of practical importance to select a model which is not only good at prediction but also cost efficient. This thesis will address the issue of cost-efficient variable selection. It was motivated by an ongoing study I was involved in, the Assertive Community Treatment (ACT) project.

1.1.1 ACT Project

The study was conducted in Southwestern Ontario to investigate Assertive Community Treatment (Lehman and Steinwachs, 1998) for patients with severe mental illness (SMI). The objectives of the study are to assess what factors influence outcomes of clients with SMI receiving care from ACT and to demonstrate the utility of a community mental health services outcomes evaluation framework. A total of 233 SMI patients were recruited. Long term outcome is the overall Colorado Client Assessment Record (CCAR) (Ellis *et al.*, 1984) Score which is revised for use in Southwest Ontario. There are about 19 potential predictive factors, such as Diagnostic Type, Total service use, Working Alliance Inventory, Empowerment, Adherence to Medication, and Present State Exam-insight Score. Data were collected from 6 sources: client self-reports, ACT clinicians, client records, hospital archive, ACT team's staff activity records, and ACT coordinators. Obviously there are great differences in the costs of collecting the data for different predictive variables or from different sources.

The cost of collecting the data has two components here. The first component is the monetary cost for human labour, time, material, equipment, compensation paid to the clients in some research activities, etc. For example, in order of decreased monetary cost, clinicians are paid for their labour to diagnose the type of mental illness of the patients and make the comprehensive treatment plan for the patients; the ACT team staffs are paid to make regular contact calls to help patients adhere to medication and make records for that; ACT patients are compensated for their time by filling out questionnaires. Another component is the level of difficulty to get an answer or a value for a potential predictor. The clients we dealt with are the patients with SMI. They may refuse to answer some questions or refuse to give certain types of information. Some results reported from the clients may need to be double checked or traced. In this sense, the client self-reported data are harder to get than the data simply obtained by chart extraction. This results in some variables being more “expensive” than others.

We consider an overall cost for each predictive factor, which is a combination of the above two components. One predictor costs more than another if this predictor is more expensive overall. Our role in the ACT project is to assess what cost-efficient factors influence outcomes of clients with SMI receiving care from ACT. We want to find the factors not only with higher prediction accuracy but are also cheaper and easier to collect the data so that we can reduce the burden of ACT team, patients, and health care system.

1.1.2 Improvement in Variable Selection Methods

Several automatic variable selection and estimation techniques have emerged in the past two decades, such as lasso (Tibshirani, 1996), boosting (Freund and Schapire, 1997) and forward stagewise regression (Hastie *et al.*, 2001). Compared with older automatic selection algorithms, such as forward selection and backward elimination, these new methods result in more parsimonious fits (simpler models are preferred) and have higher prediction accuracy (Tibshirani, 1996). The lasso (which stands for “least absolute shrinkage and selection operator”) is a popular technique for simultaneous variable selection and variable effect estimation. It achieves its goal by minimizing the residual sum of squares subject to a constraint to the sum of the absolute value of the coefficients. It shrinks some coefficients and sets others to 0 because of the constraint, which adds a little bias but reduces the variance of the predicted values, thus improving the overall prediction accuracy. Efron *et al.* (2004) introduced Least Angle Regression, abbreviated LARS (the “S” suggests “lasso” and “stagewise”). Both lasso and stagewise linear regression are variants of LARS. The key characteristic of LARS is its computational efficiency. A simple modification of the LARS algorithm implements the lasso but uses less computation time than Tibshirani’s (1996) original lasso algorithm. Zou (2006) derived a necessary condition for the lasso to be consistent and proposed a new version of the lasso, called the adaptive lasso, which adds weights in a data adaptive way to the lasso penalty term. These weights cause less shrinkage to more important predictors, which leads to consistent variable selection results.

1.1.3 The Problems We are Facing

Although the methods described in the previous section have good performance in choosing statistically important factors, they do not take into account the cost of collecting data for the predictors. Identification of cost efficient diagnostic factors is of great interest to health researchers because of the heavy burden on the public health system. Due to the development and improvement of new technology, such as nuclear medicine imaging and DNA microarray analysis, the costs of health care are escalating. In practice, inexpensive factors may have similar statistical importance as costly factors, so inexpensive factors could replace costly factors as diagnostic or prognostic variables by sacrificing minimal prediction accuracy while reducing the health cost burden. This requires statisticians to search for new strategies in statistical model building to contain the effect of the cost of collecting data for diagnostic factors. The costs may be different when collecting different variables. For example, the diabetes data used by Efron *et al.* (2004) consists of 10 variables: age, sex, body mass index (BMI), average blood pressure (BP), and S1 to S6 representing 6 serum measurements. Variables S1 to S6 have higher costs than the other variables since the data collection involves professional laboratory work, while BP and BMI have relatively lower cost due to the simpler measuring instruments, and age and sex may be assigned zero cost since they can be obtained from a registration form. For another example, collecting genetic information costs much more than collecting regular blood test results. Selecting different sets of variables in a diagnostic or prognostic model may result in “expensive” or “cheap” models, and the decision makers should make an overall judgement about the most efficient combination of variables. A model is more cost-efficient than another one if it costs less but gives almost the same prediction accuracy, or it costs much less but gives only slightly less prediction power. A health researcher, as well as a decision maker, may prefer a more cost-efficient model in many situations. If there is a budget constraint on a research project or we are at the screening stage of diagnosing a disease, a more accurate but costly model is not necessarily better than a less accurate but cheaper model.

1.2 Background and Related Work

1.2.1 Recent Automatic Variable Selection Techniques

Many variable selection and estimation techniques emerged in the past two decades, which brought great improvement both in model prediction accuracy and in computational efficiency. We give some details on these methods in the following subsections.

1.2.1.1 Least Absolute Shrinkage and Selection Operator (lasso)

Suppose we have a random sample $(\mathbf{X}, \mathbf{y})$, where $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$ is the response vector and $\mathbf{X} = (x_{ij})$, $i = 1, 2, \ldots, n$, $j = 1, \ldots, p$, is the $n \times p$ design matrix consisting of p predictors with the j^{th} predictor $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})^T$. Without loss of generality, we assume that the predictors have been standardized to have mean 0 and unit length, and that the response has mean 0.

The lasso (Tibshirani, 1996) is a constrained version of ordinary least squares (OLS). With $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^T$, the lasso estimate is defined as

$$\hat{\boldsymbol{\beta}}_{(lasso)} = \arg\min_{\boldsymbol{\beta}} \left\| \mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j \right\|^2 \quad \text{subject to } \textstyle\sum_{j=1}^{p} |\beta_j| \leq t.$$

Here $t \geq 0$ is a tuning parameter, which controls the amount of shrinkage that is applied to the estimates. lasso tends to shrink the OLS coefficients toward 0, more so for small values of t. The lasso estimate can also be written as

$$\hat{\boldsymbol{\beta}}_{(lasso)} = \arg\min_{\boldsymbol{\beta}} \left\{ \left\| \mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j \right\|^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}, \tag{1.2.1}$$

where $\lambda \geq 0$ is a regularization or tuning parameter (Efron *et al.*, 2004). The second term is often referred to as the l_1 penalty. lasso continuously shrinks the coefficients toward 0 as λ increases, and some coefficients are shrunk exactly to 0 if λ is sufficiently large. Shrinkage often improves prediction accuracy and helps to select a more parsimonious model, though there is a trade off between bias and variance. See Tibshirani (1996) and Efron *et al.* (2004) for detailed discussions.

A fast and effective way of selecting the tuning parameter λ is important in practice. The selection criteria in the literature include C_p, AIC, BIC, and Cross-validation (Hesterberg *et al.*, 2008). Efron *et al.* (2004) suggested selecting the tuning parameter and the optimal model based on C_p. Others claim that AIC is asymptotically valid if no fixed-dimension correct model exists while BIC is preferred if there exist fixed-dimension correct models (Shao, 1997; Yang, 2005). Zou *et al.* (2007) proved, without any special assumption on the predictors, that the number of nonzero coefficients is an unbiased estimate for the degrees of freedom of the lasso, which can be used in model selection criteria such as BIC. The authors discussed C_p, AIC, and BIC model selection criteria, and suggested using BIC as the model selection criterion when sparsity of the model is the major concern.

1.2.1.2 Least Angle Regression (LARS)

Efron *et al.* (2004) proposed another variable selection method referred to as LARS. Let $\hat{\boldsymbol{\mu}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ be the LARS estimates of the response, and $\hat{\boldsymbol{\beta}}$ be the estimate of the vector of coefficients. The LARS procedure works roughly as follows. Beginning at $\hat{\boldsymbol{\mu}}_0 = 0$ and setting all coefficients to zero, find the predictor, say $\mathbf{x}_1$, which is most correlated with the response. Then take the largest step possible in the direction of $\mathbf{x}_1$ (or $\mathbf{u}_1$, the unit vector along $\mathbf{x}_1$) until another predictor, say $\mathbf{x}_2$, has as much correlation with the current residual as $\mathbf{x}_1$ does. At this point, the LARS estimate is updated to $\hat{\boldsymbol{\mu}}_1 = \hat{\boldsymbol{\mu}}_0 + \hat{\gamma}_1\mathbf{u}_1$, where $\hat{\gamma}_1$ is chosen such that the current residual $\mathbf{y} - \hat{\boldsymbol{\mu}}_1$ bisects the angle between $\mathbf{x}_1$ and $\mathbf{x}_2$. Instead of continuing along $\mathbf{x}_1$, LARS proceeds in the direction of $\mathbf{u}_2$, the unit bisector of the two predictors $\mathbf{x}_1$ and $\mathbf{x}_2$, until a third variable $\mathbf{x}_3$ earns its way into the "most correlated" set. Now the LARS estimate is updated to $\hat{\boldsymbol{\mu}}_2 = \hat{\boldsymbol{\mu}}_1 + \hat{\gamma}_2\mathbf{u}_2$, where $\hat{\gamma}_2$ is chosen such that the current residual $\mathbf{y} - \hat{\boldsymbol{\mu}}_2$ has equal angles with $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$. LARS then proceeds along $\mathbf{u}_3$, the equiangular unit vector, i.e. along the "least angle direction", until a fourth variable enters, etc.

LARS builds up estimates in successive steps, each step adding one covariate to the model, so only p steps are required for the full set of solutions, where p is the number of predictors. A C_p value could be calculated for each LARS step, and Efron *et al.* (2004) suggested that the optimal LARS model is the one with the minimum C_p. LARS is computationally very efficient and it requires the same order of magnitude of computational effort as an ordinary least squares fit.

Efron *et al.* (2004) showed that lasso is a variant of LARS, so that a simple modification of the LARS algorithm could be used to implement the lasso. Using this modification, the LARS algorithm can generate the full set of lasso variable selections as λ varies. The LARS method is implemented in an R (R Development Core Team, 2008) package called *lars* (Hastie and Efron, 2007). The function *lars* in this package, by default, computes the complete lasso solution simultaneously for a sequence of values of the shrinkage parameter.

1.2.1.3 Adaptive Lasso

Lasso variable selection has been shown to be consistent under certain conditions, but there exist certain scenarios where the lasso is inconsistent for variable selection (Zou, 2006). Meinshausen and Bühlmann (2006) also showed that the optimal λ for prediction leads to inconsistent variable selection results, and many noise features are included in the prediction model. A new version of the lasso, called the adaptive lasso, could be used to fix this problem. The adaptive lasso was proposed by Zou (2006) where adaptive weights are used for penalizing different coefficients in the l_1 penalty.

Suppose $y_i = \mathbf{x}_i\boldsymbol{\beta}^* + \epsilon_i, i = 1, \ldots, n$, where $\epsilon_1, \ldots, \epsilon_n$ are independent identically distributed random variables with mean 0 and variance σ^2. The adaptive lasso estimates $\hat{\boldsymbol{\beta}}^{*(n)}$ are given by

$$\hat{\boldsymbol{\beta}}^{*(n)} = \arg\min_{\boldsymbol{\beta}} \left\{ \left\| y - \sum_{j=1}^{p} x_j \beta_j \right\|^2 + \lambda_n \sum_{j=1}^{p} \hat{w}_j |\beta_j| \right\} \tag{1.2.2}$$

The known weights vector $\hat{\mathbf{w}} = (\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_p)^T$ is data-dependent, where $\hat{w}_j, j = 1, \ldots, p$, has the form $|\hat{\beta}_j|^{-\nu}$ for $\nu > 0$ and $\hat{\beta}_j$ is a root-n-consistent estimator to β_j^*. The author suggested using the ordinary least squares estimates $\hat{\boldsymbol{\beta}}_{(ols)}$ when collinearity is not a concern. In principal, $\hat{\boldsymbol{\beta}}_{(ols)}$ can also be replaced with other consistent estimators. To find an optimal pair of (ν, λ_n), two-dimensional cross-validation can be used to tune the adaptive lasso. If $\nu = 1$ and $\hat{\mathbf{w}} = |\hat{\boldsymbol{\beta}}_{(ols)}|^{-1}$, the corresponding adaptive lasso is the same as nonnegative garrote but without the constraints $\beta_j \hat{\beta}_{j(ols)} \geq 0, \; j = 1, 2, \ldots, p$, where nonnegative garrote is consistent for variable selection (Zou, 2006). Nonnegative garrote (Breiman, 1995) is a regression procedure that minimizes

$$\sum_k \left(y_n - \sum_k c_k \hat{\beta}_k x_{kn} \right)^2$$

under the constraints

$$c_k \geq 0, \quad \sum_k c_k \leq s,$$

where $\tilde{\beta}_k(s) = c_k \hat{\beta}_k$ are the new predictor coefficients and $\{\hat{\beta}_k\}$ are the original OLS estimates. As the garrote is drawn tighter by decreasing s, more of the $\{c_k\}$ become zero and the remaining nonzero $\tilde{\beta}_k(s)$ are shrunken (Breiman, 1995).

With a proper choice of λ_n, the adaptive lasso enjoys the oracle property; namely, the adaptive lasso estimates are consistent in variable selection and are asymptotic normal $\sqrt{n}(\hat{\boldsymbol{\beta}}_{\mathcal{A}}^{*(n)} - \boldsymbol{\beta}_{\mathcal{A}}^{*(n)}) \to_d N(\mathbf{0}, \boldsymbol{\Sigma}^*)$, where $\boldsymbol{\Sigma}^*$ is the covariance matrix knowing the true subset model and $\mathcal{A} = \{j : \beta_j^* \neq 0\}$. Furthermore, the adaptive lasso can be solved by the same efficient algorithm (LARS) for solving the lasso. This method also applies to generalized linear models.

1.2.1.4 Unified Lasso Estimation via Least Squares Approximation

A method of least squares approximation (LSA) was proposed by Wang and Leng (2007) to change many different regression models into one unified theoretical framework: a unified lasso estimation. LSA can transfer many different types of lasso objective functions into their asymptotically equivalent least-squares problems. Then the standard asymptotic theory can be established and the LARS algorithm can be applied. In particular, if the adaptive lasso penalty and a BIC-type tuning parameter selector are used, the resulting LSA estimator can be very efficient.

Suppose $\mathcal{L}_n(\boldsymbol{\beta})$ is a loss function when fitting some regression model, where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T \in \Re^p$ is a parameter vector of interest. The least-squares approximation (LSA) to the loss $n^{-1}\mathcal{L}_n(\boldsymbol{\beta})$ was derived as:

$$\frac{1}{n}\mathcal{L}_n(\boldsymbol{\beta}) = (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})^T \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})$$

where $\tilde{\boldsymbol{\beta}} \in \Re^p$ is the unpenalized estimator obtained by minimizing $\mathcal{L}_n(\boldsymbol{\beta})$, and $\hat{\boldsymbol{\Sigma}}$ is a covariance matrix estimate. The original lasso problem can be rewritten as the following asymptotically equivalent least squares problem:

$$Q(\boldsymbol{\beta}) = (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})^T \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}) + \sum_{j=1}^{p} \lambda_j |\beta_j| \tag{1.2.3}$$

This objective function indicates an l_1-penalized least squares problem. It only requires the existence of a consistent covariance matrix estimate $\hat{\boldsymbol{\Sigma}}$, which is the case for most existing regression models. Thus we could change many different regression models into one unified lasso estimation and solve the above l_1-penalized least squares problem.

The LARS algorithm is an effective way to implement lasso and the R package *lars* is publicly available. Since both $\tilde{\boldsymbol{\beta}} \in \Re^p$ and $\hat{\boldsymbol{\Sigma}}$ in Equation (1.2.3) are standard outputs of many commonly used statistical packages, together with the *lars* package, the LSA method can be easily implemented for different regression models. The

LSA method uses $\tilde{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\Sigma}}$ as its standard inputs. The computation consists of one single unpenalized full model fitting for obtaining $\tilde{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\Sigma}}$, and one additional LARS processing for lasso solutions. The LSA estimator is very efficient as long as the tuning parameters are selected appropriately.

Generalized cross validation (GCV) has been extensively used to tune regularization parameters $\lambda_j, j = 1, \ldots, p$. However, if a finite dimensional model truly exists, the GCV approach tends to produce overfitted models (Wang *et al.*, 2007). A BIC-type selection criterion was suggested for tuning regularization parameter λ when using the LSA method:

$$BIC_\lambda = (\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{\beta}})^T \hat{\boldsymbol{\Sigma}}^{-1} (\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{\beta}}) + \log n \times df_\lambda / n$$

where df_λ is the number of nonzero coefficients in $\hat{\boldsymbol{\beta}}$, a simple estimate for the degrees of freedom. The optimal value of λ is the one minimizing BIC, which could then be used to find the corresponding optimal regression model.

1.2.1.5 Relaxed Lasso

For some sequences of problems where the number of predictors $p = p_n$ is growing fast with the number of observations n so that $p_n \gg n$, many noise variables are potentially selected by lasso, and lasso tends to have a low accuracy of predictions in terms of squared error loss. A two-stage procedure, named the relaxed lasso, was proposed by Meinshausen (2007), which is not only computationally efficient but also has sparser estimates and more accurate predictions.

The relaxed lasso estimator was defined as a generalization of both soft- and hard-thresholding, where model selection and shrinkage estimation are controlled by two separate parameters $\lambda \in [0, \infty)$ and $\phi \in (0, 1]$.

$$\hat{\boldsymbol{\beta}}^{\lambda,\phi} = \arg\min_{\boldsymbol{\beta}} n^{-1} \sum_{i=1}^{n} (Y_i - X_i^T \{\boldsymbol{\beta} \cdot \mathbf{1}_{\mathcal{M}_\lambda}\})^2 + \phi\lambda \|\boldsymbol{\beta}\|_1$$

where $\mathcal{M}_\lambda = \{1 \leq k \leq p \,|\, \hat{\beta}_k^{\lambda} \neq 0\}$ is the set of predictor variables selected by the lasso estimator $\hat{\beta}^\lambda$, and $\mathbf{1}_{\mathcal{M}_\lambda}$ is the indicator function on the set of variables $\mathcal{M}_\lambda$ so that for all $k \in \{1, \dots, p\}$,

$$\{\boldsymbol{\beta} \cdot \mathbf{1}_{\mathcal{M}_\lambda}\}_k = \begin{cases} 0, & k \notin \mathcal{M}_\lambda \\ \beta_k, & k \in \mathcal{M}_\lambda \end{cases}$$

Only predictor variables in the set $\mathcal{M}_\lambda$ are considered for the relaxed lasso estimator. The parameter λ controls the variable selection part, as in ordinary lasso estimation, while the relaxation parameter ϕ controls the shrinkage of coefficients. The lasso and relaxed lasso have identical estimates if $\phi = 1$, but for $\phi < 1$, the shrinkage of coefficients in the selected model using relaxed lasso is reduced compared with ordinary lasso estimation.

The simple relaxed lasso algorithm has two stages. All ordinary lasso solutions are computed at the first stage resulting a set of m models $\mathcal{M}_1, \dots, \mathcal{M}_m$ and a sequence of m penalty values $\lambda_1, \dots, \lambda_m$. At stage two, on each set $\mathcal{M}_k$ of variables, compute all lasso solutions by varying the penalty parameter between 0 and λ_k to get the set of relaxed lasso solutions $\hat{\boldsymbol{\beta}}^{\lambda,\phi}$ for $\lambda \in \Lambda_k$. The relaxed lasso solutions for all penalty parameters $\phi \in (0, 1]$ and $\lambda \geq 0$ are given by the union of these sets. The computation can be speeded up by a similar two-stage but refined algorithm.

The parameters λ, ϕ can be chosen by cross-validation which retains the fast rate. Meinshausen demonstrated that the relaxed lasso (adaptive ϕ) produces sparser models with equal or lower prediction loss than the regular lasso estimator ($\phi = 1$) for high-dimensional data.

1.2.1.6 Regularization Paths for Generalized Linear Models via Coordinate Descent

Suppose there is a response variable $y \in \Re$ and a predictor vector $\mathbf{X} \in \Re^p$, and the regression function can be approximated by a linear model $E(y|\mathbf{X} = \mathbf{x}) = \beta_0 + \mathbf{x}^T\boldsymbol{\beta}$.

There are n observation pairs $(\mathbf{x}_i, y_i), i = 1, \ldots, n$. The elastic net solves the problem

$$min_{(\beta_0,\boldsymbol{\beta})\in\Re^{p+1}} \left[R(\beta_0,\boldsymbol{\beta})\right],$$

where

$$R(\beta_0,\boldsymbol{\beta}) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \beta_0 - \mathbf{x}_i^T\boldsymbol{\beta})^2 + \lambda P_\alpha(\boldsymbol{\beta}),$$

and

$$\begin{aligned} P_\alpha(\boldsymbol{\beta}) &= (1-\alpha)\frac{1}{2}\left\|\boldsymbol{\beta}\right\|_{l_2}^2 + \alpha\left\|\boldsymbol{\beta}\right\|_{l_1} \\ &= \sum_{j=1}^{p}[\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j|] \end{aligned}$$

is the elastic-net penalty (Zou and Hastie, 2005). This penalty is particularly useful in the $p \gg n$ situation and any situation where there are many correlated predictor variables. When $\alpha = 0$, P_α is the ridge-regression penalty. It shrinks the coefficients of correlated predictors towards each other, allowing them to borrow strength from each other. When $\alpha = 1$, P_α is the lasso penalty. It shrinks many coefficients to zero. For heavily correlated predictors, it tends to pick up one of them but ignore the rest.

Generalized linear models with convex penalties, including l_1 (the lasso), l_2 (ridge regression) and mixtures of the two (the elastic net), can be estimated using a fast algorithm proposed by Friedman *et al.* (2008). The algorithm uses cyclical coordinate descent, computed along a regularization path. At each coordinate descent step, supposing that β_0 and β_l for $l \neq j$ have been estimated as $\tilde{\beta}_0$ and $\tilde{\beta}_l$, the process partially optimizes with respect to β_j. It computes the gradient at $\beta_j = \tilde{\beta}_j$, which only exists if $\tilde{\beta}_j \neq 0$. If $\tilde{\beta}_j > 0$, we have

$$\frac{\partial R}{\partial \beta_j}|_{\boldsymbol{\beta}=\tilde{\boldsymbol{\beta}}} = -\frac{1}{n}\sum_{i=1}^{n} x_{ij}(y_i - \tilde{\beta}_0 - x_i^T\tilde{\boldsymbol{\beta}}) + \lambda(1-\alpha)\beta_j + \lambda\alpha$$

A similar expression exists for $\tilde{\beta}_j < 0$. The process then coordinate-wise updates the $\tilde{\beta}_j$ based on some explicit coordinate-wise minimization formula. The authors provided several updates methods, such as naive updates, covariance updates, sparse updates, and weighted updates. Take weighted updates for example,

$$\tilde{\beta}_j \leftarrow \frac{S(\sum_{i=1}^n w_i x_{ij}(y_i - \tilde{y}_i^{(j)}), \lambda\alpha)_+}{\sum_{i=1}^n w_i x_{ij}^2 + \lambda(1-\alpha)},$$

where $\tilde{y}_i^{(j)}$ is the partial residual for fitting β_j

$$\tilde{y}_i^{(j)} = \tilde{\beta}_0 + \sum_{l \neq j} x_{il}\tilde{\beta}_l,$$

and $S(z, \gamma)$ is the soft-thresholding operator

$$\text{sign}(z)(|z| - \gamma)_+ = \begin{cases} z - \gamma, & \text{if } z > 0 \text{ and } \gamma < |z| \\ z + \gamma, & \text{if } z < 0 \text{ and } \gamma < |z| \\ 0, & \text{if } \gamma \geq |z| \end{cases}$$

After a complete cycle through all the variables, the process iterates on only the active set (nonzero coefficients) till convergence. The process stops when another complete cycle does not change the active set.

This method can work on very large datasets and can also deal efficiently with sparse features. A publicly available R package *glmnet* implements the algorithm. This coordinate-wise algorithm is considerably faster than many competing methods in some specific situations, such as LARS when $p \gg n$.

1.2.2 Cost-efficient Variable Selection

There has been relatively little work on cost efficient variable selection. To incorporate cost in a predictive model, Lindley (1968) suggested adding the cost of obtaining the covariates to the objective loss function in univariate multiple regression where a Bayesian approach was used. Brown *et al.* (1999) worked on variable selection in multivariate linear regression using a non-conjugate Bayesian decision theory approach, where a terminal cost, a function of the cost of retaining the selected variables, was included in the loss function. Their approach balances prediction accuracy against costs, and omits covariates when they cost too much relatively to their predictive benefit. We now give more details on each of these approaches.

1.2.2.1 The Choice of Variables in Multiple Regression

Lindley (1968) used a loss function

$$\{\mathbf{y} - f(\mathbf{X}_I)\}^2 + c_I,$$

where $\mathbf{y}$ is the true value of the dependent variable, I denotes a subset of the integers $1, 2, \ldots, r$ containing s members, the observed values are denoted by $\mathbf{X}_I$ whose components x_i have $i \in I$, the prediction $f(\mathbf{X}_I)$ is a function from R^s to R, and cost $c_I \geq 0$.

Squared error was used in the loss structure and an additive cost of observing the x's in I of amount c_I was assumed, that is, $c_I = \sum_{i \in I} c_i$. A Bayesian solution involves minimization of expected loss, where expectation is with respect to the probability distribution. Lindley suggested that, given the x_i's are independent, a variable is worth observing if its variation or its regression parameter is large enough compared with the cost of observation.

1.2.2.2 The Choice of Variables in Multivariate Regression

Brown *et al.* (1999) used a Bayesian decision theoretic formulation to predict an r-variate response. A non-conjugate prior distribution for the parameters of the regression model was employed. The formulation consists of a scalar loss function and a terminal cost, a function of the cost of retaining the selected p variables.

A quadratic loss was proposed as

$$\begin{aligned} \mathscr{L}(\mathbf{Y}^f, \hat{\mathbf{Y}}^f) &= (\mathbf{Y}^f - \hat{\mathbf{Y}}^f)' L (\mathbf{Y}^f - \hat{\mathbf{Y}}^f) \\ &= tr\{(\mathbf{Y}^f - \hat{\mathbf{Y}}^f)' L (\mathbf{Y}^f - \hat{\mathbf{Y}}^f)\}, \end{aligned}$$

where L is any $r \times r$ positive definite matrix of weight constants, superscript f denotes a future observation, $\hat{\mathbf{Y}}^f$ is the Bayes predictor of $\mathbf{Y}^f$ assuming all variables have been measured in the learning data $\{\mathbf{Y}^l, \mathbf{X}_q^l\}$ but that only the selection γ of the $\mathbf{X}_q^l$ is available for prediction. With $L = I$, the identity matrix, the minimized quadratic loss of the Bayes predictor could be simplified as some $tr\{R(\gamma)\}$. Then the overall loss is

$$tr\{R(\gamma)\} + g(\gamma),$$

where $g(\gamma)$ is a general terminal cost using a particular γ subset of the q regressors. The simplest form of $g(\gamma)$ is additive with a cost c_i of including variable $\mathbf{X}_i^f$ in prediction.

Predictions were judged by the quadratic loss penalized by a cost on the regressor variables. Brown *et al.* (1999) claimed that variables should be omitted not because their coefficients were believed to be zero, but because they cost too much relative to their predictive benefit.

1.2.3 Branch and Bound Search Method

An optimization problem P can be written as:

$$min\, f(\boldsymbol{\beta})$$

$$S$$

$$\boldsymbol{\beta} \in D,$$

where $\boldsymbol{\beta}$ contains the set of parameters of interest, $f(\boldsymbol{\beta})$ is the objective function, S is a set of constraints on $\boldsymbol{\beta}$, and D is the domain of $\boldsymbol{\beta}$ (Hooker, 2007). In our applications, $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)$ is a vector and D is a Cartesian product $D_1 \times D_2 \times \ldots \times D_p$, with $\beta_j \in D_j$. A feasible solution $\hat{\boldsymbol{\beta}}$ is a vector that satisfies all the constraints in S and $\hat{\boldsymbol{\beta}} \in D$. The feasible set is the set of all feasible solutions. An optimal feasible solution $\hat{\boldsymbol{\beta}}^\star$ is one with $f(\hat{\boldsymbol{\beta}}^\star) \leq f(\hat{\boldsymbol{\beta}})$ for all feasible $\hat{\boldsymbol{\beta}}$. An infeasible problem is one with no feasible solution. A problem is unbounded if there is no lower bound on $f(\hat{\boldsymbol{\beta}})$ for feasible values of $\hat{\boldsymbol{\beta}}$. An optimization problem can be either infeasible, unbounded, or have an optimal solution.

Branching search (Hooker, 2007) uses a recursive divide-and-conquer strategy that is guided by the difficulty of the problem. If the original problem P is difficult to solve, the branching algorithm branches on P and creates a series of subproblems $P_1, P_2, \ldots, P_m$. Each P_k, $k = 1, \ldots, m$ is obtained by adding constraints to the original problem P in such a way that the feasible set of P is equal to the disjoint union of the feasible sets of each P_k. Each P_k is solved, and its possible optimal solution becomes a candidate solution for P. If one P_k is too hard to solve, the search produces branches on P_k similarly. Problem P_k is said to be enumerated when it has been solved, or all of its subproblems have been enumerated. The process generates a search tree whose leaf nodes correspond to subproblems that can be either solved, shown infeasible, or shown unbounded. The optimal solution of P is the best candidate solution found.

The branch and bound search is a process that combines branching with relaxation. A relaxation R_k of P_k is created by dropping some constraints so that R_k is easier to solve than P_k. Since the feasible region of R_k contains the feasible region of P_k, the optimal value of a relaxation R_k is a lower bound on the optimal value of P_k. The branch and bound process makes use of this lower bound to accelerate the search by avoiding solution of the generally harder problem P_k. Suppose some subproblems have been solved resulting in a best candidate solution found so far, and if the optimal value of R_k, say v, is greater than or equal to the objective value of the best candidate solution found so far, then there is no need to solve P_k or branch on P_k since its optimal value can not be better than v. Problem P_k is regarded as having been enumerated even though it is not actually solved. In this case the search tree is said to be “pruned” at P_k.

1.3 Thesis Focus

Our goal in this thesis is to develop a selection procedure that can simultaneously select and estimate the important predictors to build a model that is not only good at prediction but also cost efficient. Our model is an extension of the lasso to incorporate variable costs penalized in the objective loss function. The proposed total loss includes the residual sum of squares, the lasso type penalty, and the cost of collecting data for the predictors, where the first two parts compose the lasso loss. LARS algorithm is employed since it can implement the lasso efficiently. A modified branch and bound method is proposed to search for a model which minimizes total loss. This method is referred to as the Branching LARS (BLARS) search procedure. Since the adaptive lasso can be solved by LARS using a transformation to the design matrix, we can easily adjust BLARS by using an adaptive lasso type penalty instead of a lasso type penalty, but we focus on lasso type penalty when we develop the BLARS procedure.

1.4 Thesis Organization

Introduction and other relevant background information has been highlighted in Chapter 1. We derive the theoretical basis of the BLARS method and discuss details of the implementations dealing with additive costs in Chapter 2. We address non-additive costs in Chapter 3. The software implementation, the R package *branchLars*, is displayed in Chapter 4. These chapters contain examples using the diabetes data from Efron *et al.*(2004). The BLARS method is applied to the data from the ACT project in Chapter 5. Chapter 6 provides a conclusion of materials discussed in the thesis and possible future work that can be implemented.

Chapter 2

COST-EFFICIENT VARIABLE SELECTION WITH ADDITIVE COSTS

2.1 Introduction to Additive Cost Variable Selection

The cost of collecting data for predictors cannot be avoided in the practice of risk factor identification and statistical model selection. Since the costs of health care are escalating, identification of cost efficient diagnostic factors is of great interest to health researchers. A health researcher may prefer a more cost-efficient statistical prediction model in many situations as addressed in Chapter 1.

In this chapter we are going to develop a new strategy called Branching LARS (BLARS) to take into account the effect of the cost of collecting data for diagnostic factors in statistical model building such that the model is not only good at prediction but also cost-efficient. There has been relatively little work in this area in the past.

The simplest cost structure is additive cost, which means the total cost of obtaining data for a selected set of variables is the sum of the cost of getting data for each variable in the set, i.e. $Cost_J = \sum_{j \in J} cost_j$, where $cost_j$ is the cost of collecting data for the j^{th} variable in the selected set J. The total cost cannot decrease with increased number of variables in the selected set. This cost structure applies to the situation when we collect the data for the variables individually and independently. We mainly deal with additive cost in this chapter as a starting point for its simplicity.

The rest of this chapter is organized as follows. We describe the model framework in Section 2.2, where BLARS algorithm, the order of selection of variables, and tuning parameter and model selection criteria will be addressed. Section 2.3 gives an example of data analysis using BLARS with additive cost structure. Conclusion and discussion are given in Section 2.4.

2.2 Model Framework for Additive Cost Variables

Suppose we have a random sample $(\mathbf{X}, \mathbf{y})$, where $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$ is the response vector and $\mathbf{X} = (x_{ij})$, $i = 1, 2, \ldots, n$, $j = 1, \ldots, p$, is the $n \times p$ design matrix consisting of p covariates of interest with the j^{th} predictor $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})^T$. We want to select and simultaneously estimate the coefficients of covariates such that the loss function is minimized. The total loss consists of 3 parts: the residual sum of squares of the model, the lasso type l_1 penalty, and the cost incurred by collecting data for those variables in the model. We assume that the predictors have been standardized to have mean 0 and unit length, and that the response has mean 0. With additive costs, the proposed optimization problem P can be written as

$$\min\ f(\boldsymbol{\beta}, \boldsymbol{\alpha}) = \left\| \mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j \right\|^2 + \lambda \sum_{j=1}^{p} |\beta_j| + n\gamma \sum_{j=1}^{p} \alpha_j c_j, \tag{2.2.1}$$

with domain D:

$$\begin{aligned} \alpha_j &\in \{0, 1\},\ \text{for } j = 1, \ldots, p, \\ \boldsymbol{\beta} &\in \Re^p, \end{aligned}$$

and constraints S:

$$\alpha_j = 0 \ \Rightarrow\ \beta_j = 0,\ \text{for } j = 1, \ldots, p,$$

where $\boldsymbol{\beta}$ is the regression coefficient vector that we want to estimate; $c_j \geq 0$ is the cost associated with the j^{th} variable; $\gamma \geq 0$ is a user-defined weight imposed on costs, reflecting the level of reluctance to use high cost variables; $\lambda \geq 0$ is the regularization or tuning parameter. The indicator variable α_j equals 0 or 1, indicating whether to include the cost c_j of collecting the variable $\mathbf{x}_j$. The full vector $\boldsymbol{\alpha}$ determines the variables to be selected in the regression design matrix. Note that additive costs may be unrealistic; we discuss more general cost structures in Chapter 3.

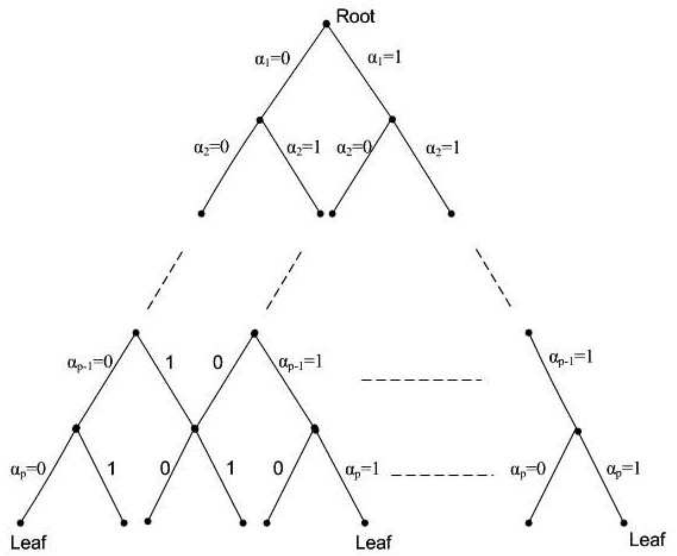

Figure 2.1: General Search Tree.

2.2.1 BLARS Method

The sum of the first two terms in the objective function (2.2.1) is the lasso objective function as in (1.2.1). The third term complicates the problem, but if we fix the value of $\boldsymbol{\alpha}$, then the third term becomes a constant and the problem reduces to lasso variable selection and estimation, and LARS may be used to solve it. A naive approach would be to try all 2^p different values of $\boldsymbol{\alpha}$, compare the results and select the best solution. Figure 2.1 displays the general search tree with all leaf-nodes corresponding to subproblems that have fixed $\boldsymbol{\alpha}$ values. In practice this approach is not feasible when p is large. The branch and bound search method can provide a solution to this problem, where relaxation is used to make the searching process easier and faster.

At each step in the BLARS process, we fix the value of one α_j to be 0 or 1. (The choice of j is discussed in section 2.2.2 below; for simplicity in this discussion we will assume numerical order, fixing α_1 first, then α_2, etc.) At step 1, we branch on the problem P and create two subproblems: $P_{1(left)}$ with $\alpha_1 = 0$ and $P_{1(right)}$ with $\alpha_1 = 1$.

We continue to branch on the subproblems and create second-level subproblems by fixing $\alpha_2 = 0$ and $\alpha_2 = 1$ respectively. Suppose at some step k, we have fixed the value of $\alpha_1, \alpha_2, \ldots, \alpha_k$, then the subproblem P_k of P has the objective (2.2.1), the same domain D and constraints S, but with the given value of $\alpha_j, j = 1, \ldots, k$. The relaxed problem R_k has the same objective (2.2.1), the same domain D and given value of $\alpha_j, j = 1, \ldots, k$, but constraints S_k:

$$\alpha_j = 0 \Rightarrow \beta_j = 0, \text{ for } j = 1, \ldots, k,$$

i.e. we drop the constraints on β_j for $j > k$. Since the feasible region of R_k contains the feasible region of P_k, the optimal objective value of the relaxed problem R_k will be a lower bound on the optimal objective value of the subproblem P_k. Without any constraints from $j = k + 1$ to $j = p$, to minimize the total loss, we set all $\alpha_j = 0$ for $j = k + 1, \ldots, p$ to solve the relaxed problem R_k, since $c_j \geq 0$.

Note that for $l < k$ and the same fixed values of $\alpha_j, j = 1, \ldots, l$, R_l is also a relaxation of R_k, so we may be able to prune certain relaxed problems, speeding up the overall search even more. Furthermore, in cases where $P_{k(right)}$ is optimized with $\beta_k = 0$, the same $\hat{\boldsymbol{\beta}}$ will optimize $P_{k(left)}$ because we force $\beta_k = 0$ for $P_{k(left)}$.

Another way to describe this search is as follows. At step k, we let $\boldsymbol{\alpha}^+$ be equal to $(\alpha_1, \ldots, \alpha_k, 1, \ldots, 1)$. This vector indicates which variables are passed to *lars* function for optimization. Once we have the *lars* result in hand, we set each of $\alpha_{k+1}, \ldots, \alpha_p$ to 0 if the corresponding $\hat{\beta}_k$ is zero, and 1 otherwise. This gives $\boldsymbol{\alpha}$ to use in the cost calculations. We also calculate $\boldsymbol{\alpha}^-$ as $(\alpha_1, \ldots, \alpha_k, 0, \ldots, 0)$ to use in the cost calculations for the bound. Figure 2.2 shows one situation of the k^{th} step of the BLARS search process.

In the algorithm below, we use the following notation. For $0 \leq k \leq p$, the solution of a relaxation R_k is denoted by $\textsc{Solution}_k$; the corresponding objective value uses $\boldsymbol{\alpha}^-$ to give the lower bound for P_k and is referred to as $\textsc{Bound}_k$. (We suppress the dependence on $\boldsymbol{\alpha}^-$, but in fact there are potentially 2^k different relaxations called

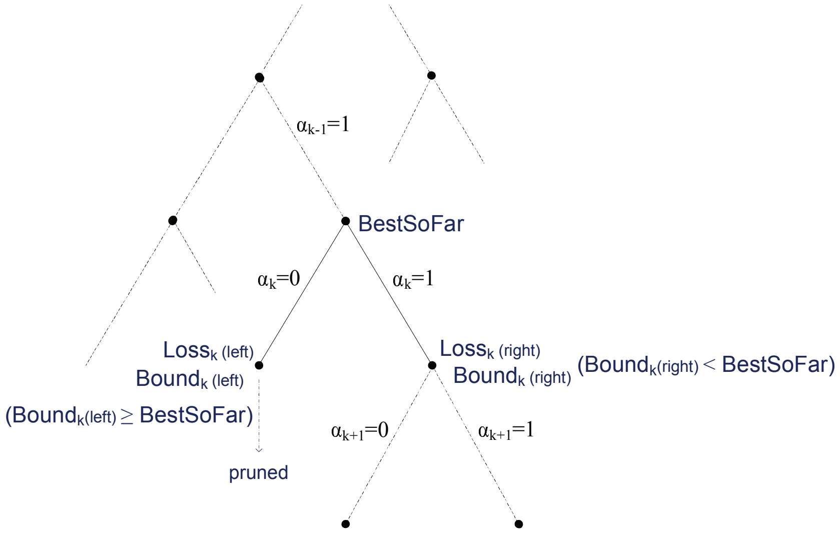

Figure 2.2: One Step of BLARS

R_k, and a corresponding number of other entities subscripted with k.) The real total loss of the model selected by R_k computed using $\boldsymbol{\alpha}$ is denoted by $\textsc{Loss}_k$. The *lars* solution from the previous step is denoted by PRESOLUTION with corresponding objective value PREBOUND. Note that $P_0 = P$, and plain *lars* is sufficient to solve R_0 since there are no restrictions on it. The best total loss seen so far is BESTLOSS.

A recursive function is called to return the best model minimizing the total loss with respect to a pre-fixed pair of tuning parameters (λ and γ). The recursive step of the BLARS algorithm is shown in Figure 2.3. This is invoked as shown in Figure 2.4.

2.2.2 The Order of Selection of Covariates

The order of the variables entering the searching process is an important factor affecting the efficiency of the algorithm. Earlier pruning will avoid searching more paths.

Intuitively, we could use the order of the LARS entries or order the variables by their costs. LARS adds one variable into the model at each step based on the

```
BLARS($k$, BestLoss, PreSolution, PreBound, $\boldsymbol{\alpha}^+$) :
    Solve $R_{k(right)}$:
        Solution$_{(right)} \leftarrow$ PreSolution
        Bound$_{(right)} \leftarrow$ PreBound $+ \gamma c_k$.
        If Bound$_{(right)} >$ BestLoss, then
            Solution$_{(right)} \leftarrow$ “pruned”
            Loss$_{(right)} \leftarrow \infty$.
            $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}^+$
        Else:
            Loss$_{(right)} \leftarrow$ Bound$_{(right)} + n\gamma \sum_{j>k:\beta_j \neq 0} c_j$.
            If Loss$_{(right)} <$ BestLoss, then
                BestLoss $\leftarrow$ Loss$_{(right)}$
            If $k < p$, then
                (Solution$_{(right)}$, Loss$_{(right)}$)
                    $\leftarrow$ BLARS($k+1$, BestLoss, Solution$_{(right)}$, Bound$_{(right)}$, $\boldsymbol{\alpha}^+$)
            Compute $\boldsymbol{\alpha}$ from Solution$_{(right)}$.
    Solve $R_{(left)}$:
        If $\alpha_k = 0$, then
            Solution$_{(left)} \leftarrow$ Solution$_{(right)}$
            Loss$_{(left)} \leftarrow$ Loss$_{(right)}$
        Else:
            $\alpha_k^+ \leftarrow 0$ and select design matrix based on $\boldsymbol{\alpha}^+$.
            Call *lars* to get Solution$_{(left)}$, Bound$_{(left)}$ and $\boldsymbol{\alpha}$.
            If Bound$_{(left)} >$ BestLoss, then
                Solution$_{(left)} \leftarrow$ “pruned”
                Loss$_{(left)} \leftarrow \infty$.
            Else:
                Loss$_{(left)} \leftarrow$ Bound$_{(left)} + n\gamma \sum_{j \geq k+1:\beta_j \neq 0} c_j$.
                If Loss$_{(left)} <$ BestLoss, then
                    BestLoss $\leftarrow$ Loss$_{(left)}$
                If $k < p$, then
                    (Solution$_{(left)}$, Loss$_{(left)}$)
                        $\leftarrow$ BLARS($k+1$, BestLoss, Solution$_{(left)}$, Bound$_{(left)}$, $\boldsymbol{\alpha}^+$)
    If Loss$_{(right)} <$ Loss$_{(left)}$, then
        Return (Solution$_{(right)}$, Loss$_{(right)}$)
    Else:
        Return (Solution$_{(left)}$, Loss$_{(left)}$)
```

Figure 2.3: The Recursive Step of the BLARS Algorithm.

```
Solve $R_0$ to find $\textsc{Solution}_0$ and $\textsc{Bound}_0$.
$\textsc{Loss}_0 \leftarrow \textsc{Bound}_0 + n\gamma \sum_{j:\beta_j \neq 0} c_j$.
Call BLARS($k = 1, \textsc{BestLoss} = \textsc{Loss}_0, \textsc{PreSolution} = \textsc{Solution}_0,$
    $\textsc{PreBound} = \textsc{Bound}_0, \boldsymbol{\alpha} = (1, \ldots, 1)$) and return the result.
```

Figure 2.4: The Initialization Step of the BLARS Algorithm.

correlation of the variables with the updated response. The most correlated covariate is added first and the least correlated one is the last. If we let the most correlated covariate enter the BLARS searching process first, the lasso loss (the first two terms in the objective function (2.2.1)) may decrease dramatically, and the tree is more likely to be pruned at the node where we force this variable out of the model, i.e. the node where we let $\alpha_1 = 0$. Using this ordering method, the computing time may be reduced because the tree has more chance to be pruned at upper level left-path nodes. On the other hand, when the cost difference of the predictors is large (usually associated with a higher value of γ), the cost effect may dominate. Ordering variables by descending order of the costs could be a better approach in this case. If we let the most expensive covariate enter the BLARS searching process first, the gain by the decrease of the lasso loss may be clearly surpassed by the loss by the increase of the cost, and the tree is more likely to be pruned at the node where we force this variable in the model, i.e. the node where we let $\alpha_1 = 1$. Using this ordering method, the computing time may be reduced because the tree has more chance to be pruned at upper level right-path nodes.

To get a quantitative analysis, we compared 6 ordering methods by assessing how many times the search method called the *lars* function in the searching process for different combinations of λ and γ values using the diabetes data (used by Efron *et al.* 2004). The 6 methods are to order the potential covariates in descending order of the correlations with the updated response, i.e. the order of the LARS entries (*Lars Descending*), ascending order of the correlations (*Lars Ascending*), descending order of the costs (*Cost Descending*), ascending order of the costs (*Cost Ascending*), descending order of the absolute value of the OLS estimates (*OLS Descending*) and ascending order of the absolute value of the OLS estimates (*OLS Ascending*) respectively. We change the order at the beginning of the searching process and only once when using the order of *Cost Descending*, *Cost Ascending*, *OLS Descending* or *OLS Ascending*. For the order of *Lars Descending* or *Lars Ascending*, we may change the order of the variables more than once. At the beginning, we change the variables in descending or ascending order of the LARS entries based on an initial *lars* call. Since we need to call the *lars* function many times in the searching process, the order of the LARS entries may alter after a later *lars* call. When this happens, we change the order of the variables accordingly to further reduce the searching time. Figure 2.5 shows the number of *lars* calls for 4 different λ values. Corresponding to different range of γ, either the *Lars Descending* or the *Cost Descending* ordering method has the best result.

We suggest using the order of the LARS entries when γ is relatively small or equivalently the cost difference of the covariates is not too big, and when γ is large but λ is also relatively large; and use the descending order of the cost for other situations. In practice, we may need to fit BLARS for multiple γ values. In this case we should start with small γ and use the order of the LARS entries; subsequently we can occasionally try both ordering methods to make a comparison and switch to the descending order of the cost at the point where the cost method is the better choice.

2.2.3 Tuning Parameter and Model Selection Criteria

The parameter γ is a user-defined weight imposed on costs, reflecting the level of reluctance to use high cost variables. When $\gamma = 0$, we ignore the costs and selection becomes the standard lasso variable selection. The higher the γ value, the more

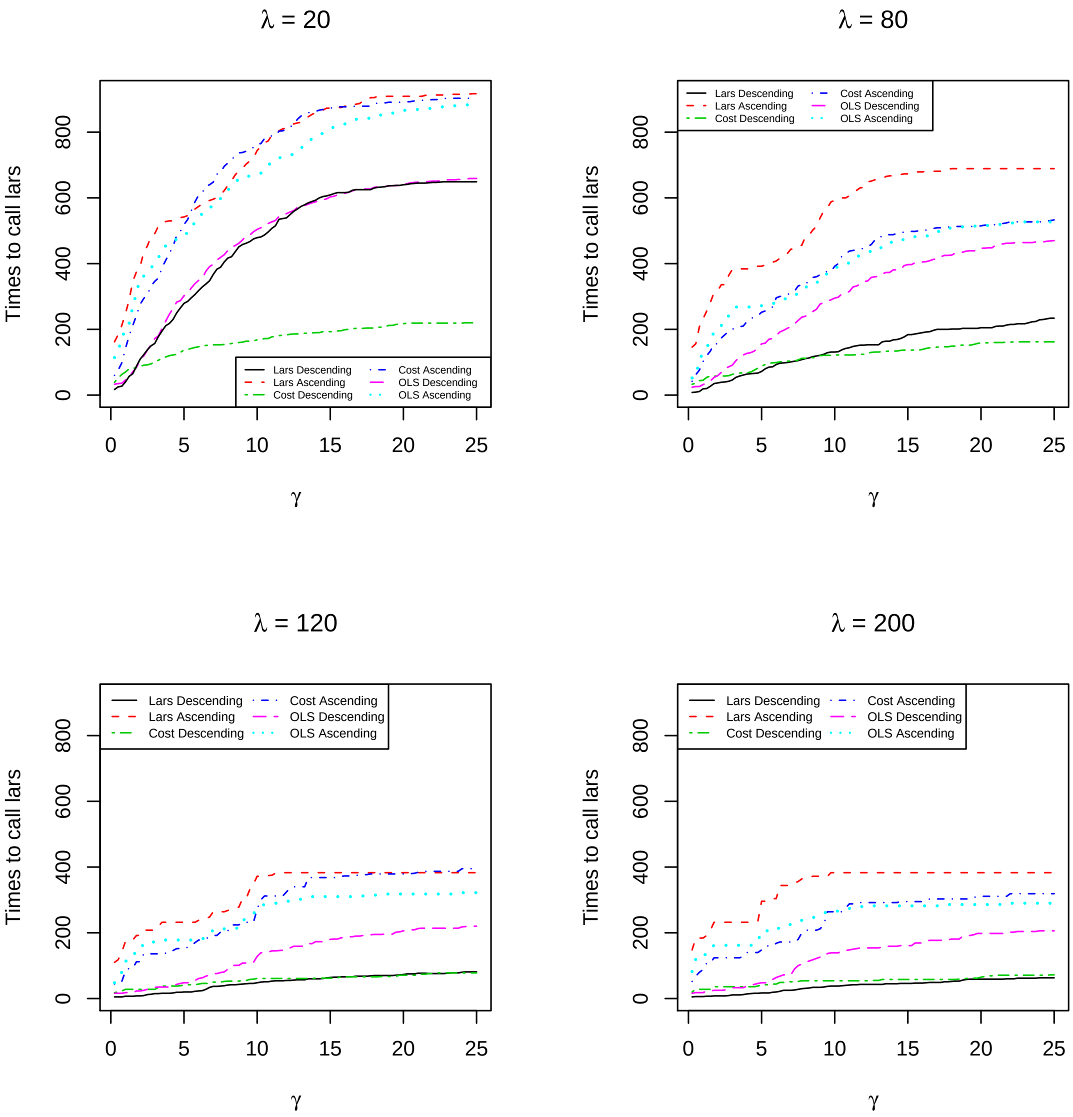

Figure 2.5: Computing times for different ordering methods.

reluctant is the user to select high cost variables. Thus when the user assigns a higher value to γ, the BLARS process will be less likely to select higher cost variables. The assignment of a γ value is thus based to a large extent on the opinions and judgments of the user or the decision maker. Sometimes the user has to use a higher γ because of budget constraints. Once γ is fixed, the optimal value of λ and the corresponding optimal statistical model could be selected by any model selection criterion. In the example in Section 2.3, we use BIC for the lasso, which was proposed by Zou *et al.* (2007), as the tuning parameter and model selection criterion for its simplicity and effectiveness.

$$BIC(\hat{\boldsymbol{\mu}}) = \frac{\|\mathbf{y} - \hat{\boldsymbol{\mu}}\|^2}{n\sigma^2} + \frac{log(n)}{n}\hat{df}(\hat{\boldsymbol{\mu}}) \tag{2.2.2}$$

where $\hat{df}(\hat{\boldsymbol{\mu}})$ equals the number of nonzero coefficients, which is an unbiased estimate for the degrees of freedom of the lasso showed by Zou *et al.* (2007).

2.2.4 Pruning Based on Previous Fits

Proposition 1. Given a fixed value of λ in the BLARS minimization procedure, the value of $\sum_{j=1}^{p} \alpha_j c_j$ in the optimal model cannot increase when we increase the value of γ.

Proof. The first two terms in (2.2.1) depend only on $\boldsymbol{\beta}$, while the last term depends only on $\boldsymbol{\alpha}$. Since $\gamma > 0$, an increase in $\sum_{j=1}^{p} \alpha_j c_j$ would imply a decrease in the first two terms, implying a better solution at the original γ value. ■

In practice, when we fix a value of λ and increase the value of γ from γ_1 to γ_2, we could ignore all those variables that have higher cost than the sum of the costs of the variables selected using γ_1 in the BLARS search process. More generally, we may prune a branch simply because $\sum_{j=1}^{p} \alpha_j c_j$ of this branch is larger than the one in the optimal model for γ_1.

Table 2.1: Additive Cost of Variables Used in Example.

Predictors	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6
Costs	0	0	5	10	50	50	50	50	50	50

2.3 An Example for Additive Cost Variable Selection

We apply the BLARS algorithm to the diabetes data used by Efron *et al.* (2004). The data consists of 10 variables: age, sex, body mass index (BMI), average blood pressure (BP), and S1 to S6 representing 6 serum measurements.

The cost of collecting a variable may include the cost of material, equipment, time, human labour, etc. One way to assign a cost would be to use the dollar amount we have to pay to get that variable; a more sophisticated analysis might include the time required or the discomfort of the test. For the purpose of illustration, we assign the costs to the 10 variables as shown in Table 2.1. S1 to S6 have the highest costs since the collection involves professional laboratory work; BP and BMI have relatively lower cost due to the simpler measuring instruments; while age and sex are assigned zero cost since they may be obtained from a registration form.

2.3.1 Cost Efficient Variable Selection with Additive Costs

Table 2.2 displays the variable selection results for some combinations of λ and γ. The corresponding order of the variables in the BLARS search process is showed in Table 2.3. When $\gamma = 0$, the BLARS results are equivalent to the lasso results. For a fixed value of λ, BLARS tends to select fewer or cheaper variables as γ increases. For a fixed value of γ, BLARS tends to select fewer variables as λ increases, as does the usual lasso selection method. Figure 2.6 shows the search tree when $\lambda = 20$ and $\gamma = 0.3$. Values of α_k are shown for each variable. The red path is the chosen optimal

Table 2.2: Coefficients Resulting from Cost-efficient Variable Selection with Additive Cost.

		Estimated Coefficients									
λ	γ	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6
20	0	0	-18.85	5.63	1.02	-0.14	0	-0.82	0	46.92	0.23
	0.3	0	-18.39	5.72	1.06	-0.13	0	-0.84	0	47.99	0
	1	0	-19.02	5.61	1.05	0	0	-0.97	0	42.75	0
	7	-0.02	-10.41	6.34	0.95	0	0	0	0	49.45	0
	15	0.07	-7.50	8.34	1.38	0	0	0	0	0	0
	30	0.43	-1.04	9.79	0	0	0	0	0	0	0
200	0	0	0	5.16	0.51	0	0	-0.26	0	37.86	0
	0.3	0	0	5.16	0.51	0	0	-0.26	0	37.86	0
	1	0	0	5.35	0.50	0	0	0	0	39.83	0
	7	0	0	5.79	0	0	0	0	0	43.45	0
	15	0	0	6.97	0.89	0	0	0	0	0	0
	30	0	0	8.08	0	0	0	0	0	0	0

path.

Corresponding to Table 2.2, the error sum of squares (SSE), cost, and their percentage contribution to the total loss are shown in Table 2.4. For fixed λ and increasing γ, at first the percentage of SSE in total loss is decreasing, but as the percentage of cost increases, fewer and cheaper variables are selected and the percentage of SSE in the total loss increases again.

We show the computational efficiency of BLARS in Tables 2.5 and 2.6. There are 10 potential predictors, hence $2^{10} = 1024$ different $\boldsymbol{\alpha}$ values. If no variable is selected with an initial *lars* call for some λ values, the search process stops there. For

Table 2.3: Initial order of variables in BLARS.

		Ordering										
λ	γ	Method	Order of entry from first to last									
20	0.3	lars	BMI	S5	BP	S3	SEX	S6	S1	S4	S2	AGE
	1	lars	BMI	S5	BP	S3	SEX	S6	S1	S2	S4	AGE
	7	lars	BMI	S5	BP	S3	S4	SEX	S1	S6	S2	AGE
	15	lars	BMI	S5	BP	S4	S3	S6	SEX	S1	AGE	S2
	30	cost	S1	S2	S3	S4	S5	S6	BP	BMI	AGE	SEX
200	0.3	lars	BMI	S5	BP	S3	SEX	S6	S1	S4	S2	AGE
	1	lars	BMI	S5	BP	S3	S4	SEX	S1	S6	S2	AGE
	7	lars	BMI	S5	BP	S3	S6	S4	S1	SEX	AGE	S2
	15	lars	BMI	S5	BP	S4	S3	S6	SEX	S1	AGE	S2
	30	lars	BMI	S5	BP	S4	S3	S6	AGE	S1	S2	SEX

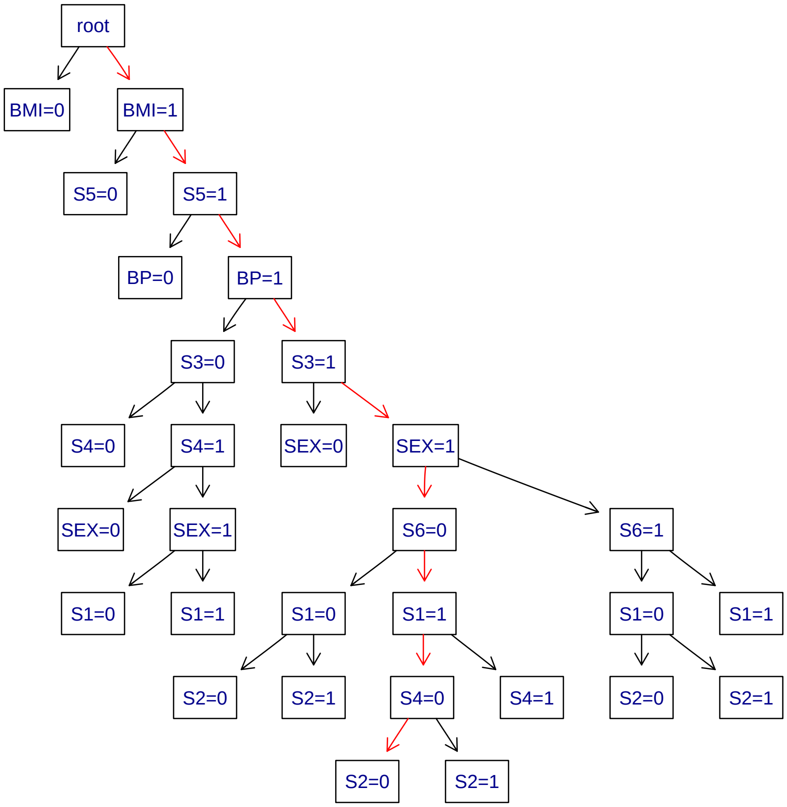

Figure 2.6: Search Tree for $\lambda = 20$ and $\gamma = 0.3$ with Additive Cost.

Table 2.4: Effect of γ on SSE and Additive Cost.

		SSE	Percentage of SSE	Cost ($n\gamma \sum \alpha_j c_j$)	Percentage of Cost
λ	γ	(thousand)	in total loss	(thousand)	in total loss
20	0	1275	97.09%	0	0%
	0.3	1279	95.55%	22	1.63%
	1	1290	93.75%	51	3.69%
	7	1346	85.32%	201	12.75%
	15	1573	92.64%	99	5.86%
	30	1703	95.14%	66	3.70%
200	0	1411	86.35%	0	0%
	0.3	1411	85.55%	15	0.92%
	1	1429	85.37%	29	1.72%
	7	1472	79.79%	170	9.22%
	15	1640	85.39%	99	5.18%
	30	1760	89.06%	66	3.36%

Table 2.5: Computational Efficiency of BLARS with Additive Cost, for Fixed Choices of λ and γ.

λ	γ	Number of Times to Call *lars*	Ordering Method	Proportions of Full Search (Times/1023)
20	0.3	16	lars	1.56%
	1	20	lars	1.96%
	7	34	lars	3.32%
	15	46	lars	4.50%
	30	63	cost	6.16%
200	0.3	5	lars	0.49%
	1	6	lars	0.59%
	7	14	lars	1.37%
	15	17	lars	1.66%
	30	18	lars	1.76%

general situations, we would need to call the *lars* function up to 1023 times to make a full search (there is no need to call *lars* when all $\alpha_j = 0$.) When using the BLARS approach, the number of times *lars* is called is much reduced. For fixed γ, computing time decreases with λ, as simpler models are selected. For fixed λ, computing time increases with γ. When λ is small and γ is large, we may order the variables by cost to reduce the computing time. With fixed λ, Proposition 1 allows us to use the results of fits with small γ for additional efficiencies (Table 2.6).

2.3.2 Optimal Models Using Additive Costs

We choose the optimal tuning parameter and optimal model based on the BIC for lasso (Zou *et al.*, 2007). Figure 2.7 displays the BIC values for models with different

Table 2.6: Computational Efficiency of BLARS with Additive Cost Using Smaller γ Values to Save Search Time for Larger Ones.

		Number of Times	Ordering	Proportions of Full Search
λ	γ	to Call *lars*	Method	(Times/1023)
20	0.3	16	lars	1.56%
	1	20	lars	1.96%
	7	29	lars	2.83%
	15	34	lars	3.32%
	30	10	cost	0.98%
200	0.3	5	lars	0.49%
	1	6	lars	0.59%
	7	14	lars	1.37%
	15	15	lars	1.47%
	30	13	lars	1.27%

Table 2.7: Optimal Models Chosen by Lasso and BLARS with Additive Cost.

	Estimated Coefficients									
Method	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6
Lasso	0	-18.85	5.63	1.02	-0.14	0	-0.82	0	46.92	0.23
BLARS	0	-20.88	5.63	1.09	0	0	-1.02	0	43.01	0

λ and γ. For each λ, BIC is a step function of γ. Each jump in BIC corresponds to one model change, while BIC remains the same when selected variables are the same but γ is increasing. The minimum BIC corresponds to a low level of λ and γ, which is displayed more clearly in Figure 2.8. BIC is decreasing at the beginning with increasing γ for a fixed λ; it reaches the minimum for a range of γ values and then increases. The optimal value of λ based on BIC depends on the value of γ. For this data set, if we are free to choose γ, the optimal λ should be around 10 and the minimum BIC should be below 1.065 based on Figure 2.8, with the γ value ranging roughly from 0.5 to 2.4. This optimal solution was found to be $\lambda = 9.22$ and $BIC = 1.063$. We compare the corresponding optimal model (only for γ in the rough range from 0.5 to 2.4) with the model selected by the lasso based on BIC criterion in Table 2.7. The optimal model using BLARS contains two less predictors: S1 and S6, which is due to the cost effect. The total cost is decreased by 46.5%, and this "cheaper" model is selected by sacrificing a 1.01% increase of SSE.

2.4 Conclusion and Discussion

The BLARS method for cost-efficient variable selection with additive cost structure has been developed in this chapter, and it has been shown that, through assigning a γ value, a "cheaper" model could be selected by sacrificing a user selected amount of

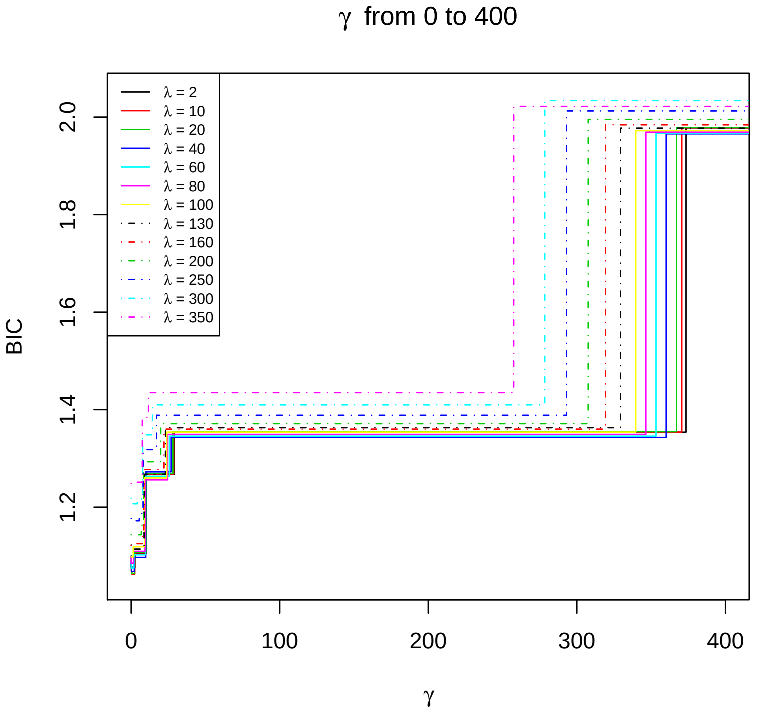

Figure 2.7: BIC values for different λ and γ.

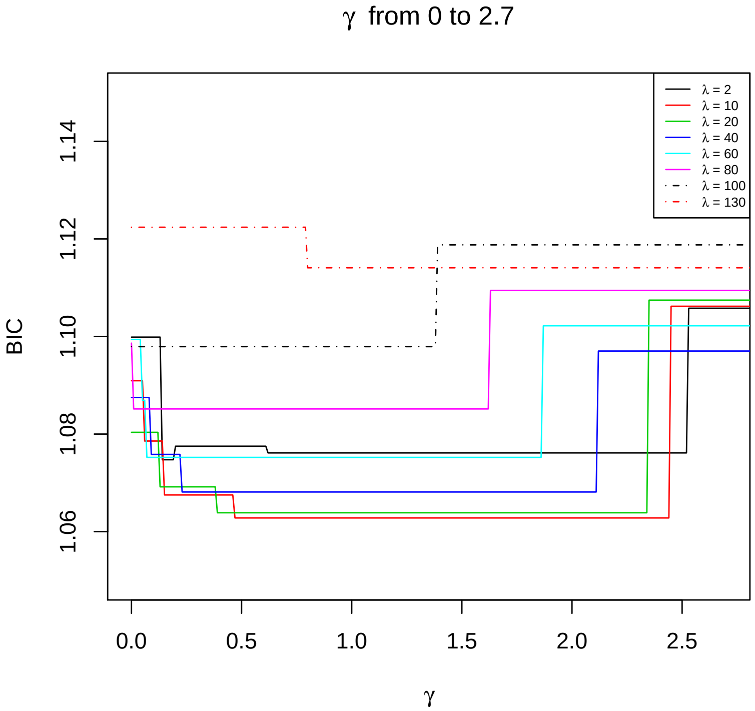

Figure 2.8: BIC values for small λ and γ.

model accuracy.

We solved a discrete optimization problem (2.2.1) where the indicator variable α_j equals 0 or 1. Naively, one might think that the optimization problem could be approached with a conventional method which assumes that the α_j can take on any value between 0 and 1; once the estimates are obtained, they could be rounded to 0 or 1. Unfortunately, such an approach will fail because the constraints in (2.2.1) introduce a discontinuity at $\alpha_j = 0$. Therefore, a discrete optimization method such as branch and bound is necessary.

We have used an additive cost in this chapter. However, additive costs may be unrealistic. For example, there may be grouping effects, e.g. when one variable is selected some other variables become free or cheaper, such as in microarray gene expression data analysis where gene expression data are obtained from one experiment. Another grouping cost may come from the situation where higher order or interaction terms are considered in a model: these cost nothing once the variables have been measured. We will address more general cost structure in the next chapter.

Chapter 3

COST-EFFICIENT VARIABLE SELECTION WITH NON-ADDITIVE COSTS

3.1 Introduction to Non-additive Cost Variable Selection

We have developed the BLARS cost-efficient variable selection method in Chapter 2 which exclusively deals with an additive cost structure. However, additive cost may not be realistic in practice since the data for the predictors are seldom collected fully independently. For instance, when we collect a number of blood test results, some results are obtained using the same blood sample. In this situation, there is a cost for collecting the blood sample, and then an additional cost for each blood test result involving different laboratory work. The total cost for obtaining these blood test results is not simply the sum of the individual costs as in Chapter 2, but is the sum of two parts: the unique cost of the blood sample and the sum of additional cost for each blood test result.

The more general cost structure is a non-additive cost, such as the grouping effect of costs. For example, blood test results were obtained using the same blood sample. Suppose the cost of collecting the blood sample is C_1 (fixed cost for group) and the additional cost for each test result is C_{2j} (marginal cost), where $j = 1, \ldots, p$ and p is the total number of blood test results in this group. The cost for each test result is $C_1 + C_{2j}$ before any of them being selected into the BLARS model. When one test result (say x_1) is included in the BLARS model, the other test results become cheaper (the cost is only C_{2j} instead of $C_1 + C_{2j}$ for $j = 2, \ldots, p$) if they are also selected in the BLARS model. Another grouping effect occurs when higher order or interaction terms are considered in a model. These terms become free once the variables involved in the terms have been selected. We will deal with this more general non-additive cost in this chapter.

The rest of this chapter is organized as follows. We describe the model framework for non-additive cost variables in Section 3.2, where the improved BLARS process, the combined order of covariates in selection, and non-additive cost structure will be addressed. An example of data analysis using BLARS with non-additive cost structure is showed in Section 3.3. Section 3.4 gives conclusion and discussion.

3.2 Model Framework for Non-additive Cost Variables

Similarly as in Chapter 2, suppose we have a random sample $(\mathbf{X}, \mathbf{y})$, where $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$ is the response vector and $\mathbf{X} = (x_{ij})$, $i = 1, 2, \ldots, n$, $j = 1, \ldots, p$, is the $n \times p$ design matrix consisting of p covariates of interest with the j^{th} predictor $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})^T$. By location and scale transformation we assume that the predictors have been standardized to have mean 0 and unit length, and that the response has mean 0. We want to select and simultaneously estimate the coefficients of covariates such that the loss function is minimized. With non-additive costs, the proposed optimization problem P can be written as

$$min\ f(\boldsymbol{\beta}, \boldsymbol{\alpha}) = \left\| \mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j \right\|^2 + \lambda \sum_{j=1}^{p} |\beta_j| + n\gamma C(\alpha_1 c_1, \ldots, \alpha_p c_p), \qquad (3.2.1)$$

with domain D:

$$\begin{aligned} \alpha_j &\in \{0, 1\},\ for\ j = 1, \ldots, p, \\ \boldsymbol{\beta} &\in \Re^p, \end{aligned}$$

and constraints S:

$$\alpha_j = 0 \ \Rightarrow\ \beta_j = 0,\ for\ j = 1, \ldots, p,$$

where $\boldsymbol{\beta}$ is the regression coefficient vector that we want to estimate; $\lambda \geq 0$ is the regularization or tuning parameter; $\gamma \geq 0$ is a user-defined weight imposed on costs, reflecting the level of reluctance to use high cost variables. The p-vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)$ contains 0's and 1's. Thus, every α_j is an indicator variable. The cost c_j of collecting the variable $\mathbf{x}_j$ is included in the total cost C if $\alpha_j = 1$ but is not included if $\alpha_j = 0$. The cost function $C(\alpha_1 c_1, \ldots, \alpha_p c_p)$ is non-decreasing when adding more $\alpha_k = 1$ to the existing non-zero set of $\{\alpha_j\}$. The constraints indicate that $\boldsymbol{\alpha}$ is used to determine the variables to be selected in the regression design matrix.

3.2.1 Improved BLARS Process

The BLARS method combines LARS with branch and bound search method, where relaxation is used to make the searching process easier and faster. Although BLARS was developed based on an additive cost, the algorithm is also suitable to deal with non-additive cost, and it will be usable whenever costs are a non-decreasing function of the set of selected variables.

When dealing with non-additive cost, we make some changes and improvements for the BLARS process described in Chapter 2. First, after each step in the BLARS process, we reassess the cost of each potential predictor based on variables selected in the model in the previous steps, since the costs of the remaining undetermined variables may be changed due to the grouping effect of the costs. Second, we use a combined ordering method to order the variables entering the searching process to make the BLARS process more computationally efficient. This ordering method combines the *LARS* entries with the *Cost* ordering method, and the details are showed in Section 3.2.2. Third, since many variables may become free due to the grouping effect of the costs and these variables no longer have the cost effect we are concerning about, we put these zero-cost variables in the last positions to enter the BLARS searching process, and actually we stop the search when all the remaining undetermined variables have zero-cost. Note that different zero-cost variables may appear at some BLARS step since different sets of variables are selected at different nodes of the search tree. Furthermore, based on Proposition 2, we may prune a branch if the value of C of this branch is larger than the one in the optimal model for a smaller γ.

Proposition 2. Given a fixed value of λ in the BLARS minimization procedure, the value of $C(\alpha_1 c_1, \ldots, \alpha_p c_p)$ in the optimal model cannot increase when we increase the value of γ.

Proof. The first two terms in (3.2.1) depend only on $\boldsymbol{\beta}$, while the last term depends only on $\boldsymbol{\alpha}$. Since $\gamma > 0$, an increase in $C(\alpha_1 c_1, \ldots, \alpha_p c_p)$ would imply a decrease in the first two terms, implying a better solution at the original γ value. ■

3.2.2 The Combined Order of Covariates in Selection

The order of the variables entering the searching process is an important factor affecting the efficiency of the selection program. Earlier pruning will avoid searching more paths.

In Chapter 2, we compared several ordering methods by assessing how many times the search method calls the *lars* function in the searching process for different combinations of λ and γ values using the diabetes data (used by Efron *et al.* 2004). We found that corresponding to different range of γ, ordering the potential covariates either in descending order of the correlations with the updated response, i.e. the order of the LARS entries (*LARS*) or descending order of the costs (*Cost*) has the best result.

We combine the *LARS* with the *Cost* ordering method in this chapter to make the search process more efficient. First we calculate a bin value for the costs of the potential predictors. The proposed formula is:

$$bin = 0.25 \frac{1}{\gamma} \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2,$$

where γ is the user-defined weight imposed on the costs, and $\bar{y}$ is the mean value of the response $\mathbf{y}$. Then the cost of each potential predictor falls into one range: $k \times bin \leq c_j < (k+1) \times bin$, where $k \geq 0$ is an integer and $j = 1, \ldots, p$. This is shown in Figure 3.1. We order the variables in different bins by the *Cost* method, while order the variables in the same bin by the *LARS* method. Thus for the case in Figure 3.1, x_5 and x_6 are the first two variables entering the BLARS search process since they have the highest costs, but which one is the first depends on the LARS entries. The variables in the lowest bin, such as x_1, x_2, x_3 in Figure 3.1, are the last ones entering the BLARS search process, and we further order these three variables based on the LARS entries; however, we always put the variables with 0 cost in the last positions and we do not even need to search through them due to the zero-cost.

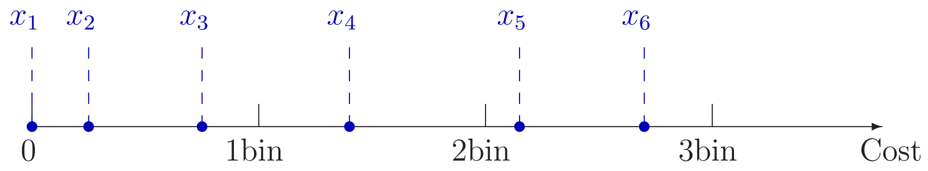

Figure 3.1: The Bin of the Costs

3.2.3 Non-additive cost

We used an additive cost in the previous chapter. However, there may be non-additive cost, such as grouping effects of cost. Figure 3.2 shows the grouping effect of the costs. Take group 1 as an example. Suppose x_1, x_3, x_4 belongs to group 1 with group cost C_1 and individual additional cost $0, C_3, 0$ respectively. If x_1 is selected, x_4 becomes free and x_3 is cheaper (cost only C_3) if they are also selected. The total cost is $C_1 + C_3$ if all the variables in group 1 are selected but C_1 if only x_1 and x_4 are selected. Another grouping cost may come from the situation where higher order or interaction terms are considered in a model. For instance, if x_1 and x_2 have been selected into the BLARS model, then the squared terms x_1^2 and x_2^2 and the interaction term $x_1 : x_2$ become free it they are selected. It is clear that BLARS will be usable whenever costs are non-decreasing function of the set of included variables.

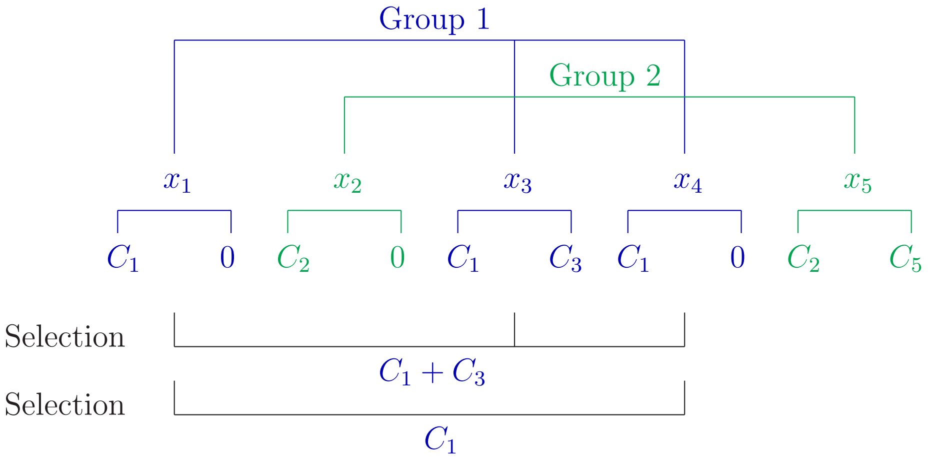

Figure 3.2: The Grouping Effect of the Costs

In BLARS, we deal with non-additive cost by updating the cost of each of the undetermined variables (the variables that have not entered the search process) after each step based on which variables have been selected into the model. For example, the cost for each variable in Figure 3.2 is $C_1, C_2, C_1 + C_3, C_1, C_2 + C_5$ for x_1 to x_5 respectively before they enter the search process. Suppose after a number of steps, x_1, x_2 and x_3 have entered the search process and x_1 and x_2 have been selected into the model, then we update the cost for x_4 and x_5 to be 0 and C_5 respectively. Each time after we update the costs for the undetermined variables we need to reorder them based on their new costs using the Cost-combine-LARS ordering method described in section 3.2.2.

3.3 An Example for Non-additive Cost Variable Selection

We apply the BLARS to the diabetes data used by Efron *et al.* Efron *et al.* (2004). The data consists of 10 variables: age, sex, body mass index (BMI), average blood pressure (BP), and S1 to S6 representing 6 serum measurements. Suppose the 6 blood test results are obtained using the same blood sample. For the purpose of illustration, we assign the non-additive costs to the 10 variables as shown in Table 3.1.

Table 3.1: Non-additive Cost of Variables Used in Example.

Predictors	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6
Group Number	1		2	3	4					
Group Cost	0		5	10	20					
Additional Costs	0	0	0	0	30	30	30	30	30	30

3.3.1 Cost Efficient Variable Selection with Non-additive Costs

We choose the tuning parameter $\lambda = 20$ and $\lambda = 200$ as examples to illustrate the cost efficient variable selection process. γ is a user-defined weight imposed on costs, reflecting the level of reluctance to use high cost variables. Table 3.2 displays the variable selection results for some combinations of λ and γ. When $\gamma = 0$, the BLARS results are equivalent to the standard lasso results. For a fixed value of λ, BLARS tends to select fewer or cheaper variables as γ increases. For a fixed value of γ, BLARS tends to select fewer variables as λ increases, as does the usual lasso selection method. Figure 3.3 shows the search tree when $\lambda = 20$ and $\gamma = 0.5$ with non-additive cost. Values of α_k are shown for each variable. The red path is the chosen optimal path.

Corresponding to Table 3.2, the effect of γ on the error sum of squares (SSE) and cost are shown in Table 3.3. For fixed λ and increasing γ, at first the percentage

Table 3.2: Coefficients Resulting from Cost-efficient Variable Selection with Non-additive Cost.

		Estimated Coefficients									
λ	γ	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6
20	0	0	-18.85	5.63	1.02	-0.14	0	-0.82	0	46.92	0.23
	0.5	0	-18.39	5.72	1.06	-0.13	0	-0.84	0	47.99	0
	1	0	-19.02	5.61	1.05	0	0	-0.97	0	42.75	0
	7	-0.02	-10.41	6.34	0.95	0	0	0	0	49.45	0
	15	0.07	-7.50	8.34	1.38	0	0	0	0	0	0
	35	0.43	-1.04	9.79	0	0	0	0	0	0	0
200	0	0	0	5.16	0.51	0	0	-0.26	0	37.86	0
	0.5	0	0	5.16	0.51	0	0	-0.26	0	37.86	0
	1	0	0	5.35	0.50	0	0	0	0	39.83	0
	7	0	0	5.79	0	0	0	0	0	43.45	0
	15	0	0	6.97	0.89	0	0	0	0	0	0
	35	0	0	8.08	0	0	0	0	0	0	0

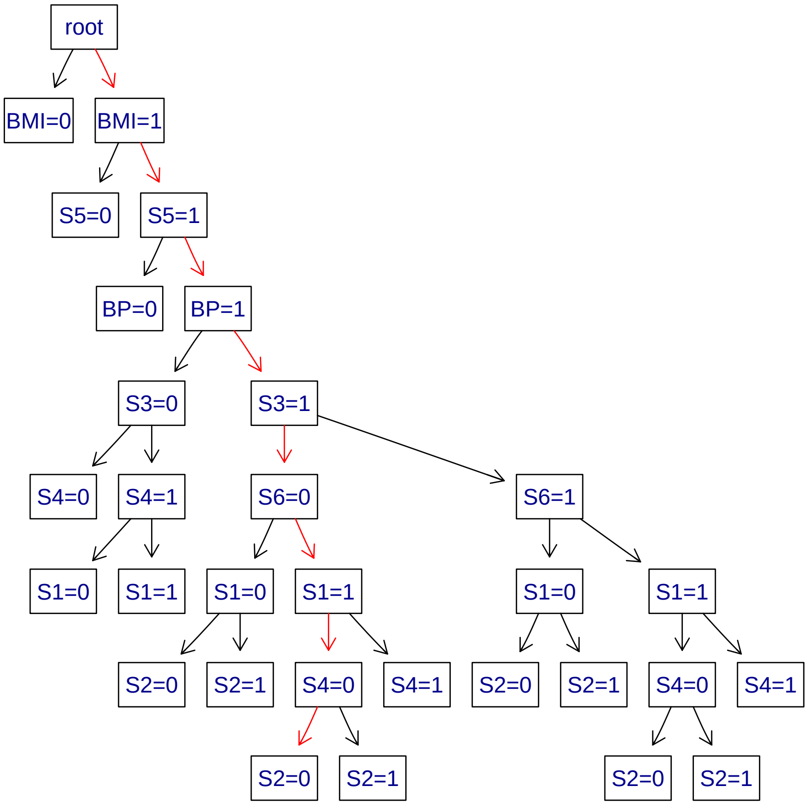

Figure 3.3: Search Tree for $\lambda = 20$ and $\gamma = 0.5$ with Non-additive Cost.

of SSE in total loss is decreasing, but as the percentage of cost increases, fewer and cheaper variables are selected and the percentage of SSE in the total losses increases again.

Table 3.3: Effect of γ on SSE and Non-additive Cost.

		SSE	Percentage of SSE	Cost ($n\gamma C$)	Percentage of Cost
λ	γ	(thousand)	in total loss	(thousand)	in total loss
20	0	1275	97.09%	0	0%
	0.5	1279	95.14%	28	2.06%
	1	1290	94.36%	42	3.07%
	7	1346	85.32%	201	12.75%
	15	1573	92.64%	99	5.86%
	35	1703	94.55%	77	4.30%
200	0	1411	86.35%	0	0%
	0.5	1411	85.26%	21	1.27%
	1	1429	85.37%	29	1.72%
	7	1472	79.79%	170	9.22%
	15	1640	85.39%	99	5.18%
	35	1760	88.56%	77	3.89%

The computational efficiency of BLARS dealing with non-additive cost are showed in Tables 3.4 and 3.5. There are 10 potential predictors, we usually need to call the *lars* function up to 1023 times to make a full search for general situations. When using the BLARS approach, the number of times the *lars* is called is much reduced. With fixed λ, we could take the sum of costs of previously selected variables using a lower γ as an input value for a new BLARS call with a higher γ to reduce the computing time (Table 3.5). Another time-saving approach is to ignore some variables that have a higher cost than the sum of the costs of the variables selected previously using a lower γ. For instance, when $\lambda = 20$ and $\gamma = 15$, the process selects the first four variables and the sum of the costs for each patient is 15; if we increase γ, we could delete the variables S1 to S6 (have cost ≥ 30) from the design matrix and still get the same results, but the times to call *lars* function drops dramatically (only 3 times compared with the previous 38 when $\gamma = 35$ in Table 3.4).

Table 3.4: Computational Efficiency of BLARS with Non-additive Cost, for Fixed Choices of λ and γ.

		Number of Times	Proportions of Full Search
λ	γ	to Call *lars*	(Times/1023)
20	0.5	16	1.56%
	1	19	1.86%
	7	31	3.03%
	15	39	3.81%
	35	38	3.71%
200	0.5	9	0.88%
	1	13	1.27%
	7	23	2.25%
	15	27	2.64%
	35	25	2.44%

3.3.2 Full Model

We could include squared terms and two-way interaction terms in the design matrix. Excluding the SEX^2 term, we have 64 potential predictors.

We choose the tuning parameter $\lambda = 60$ as examples to illustrate the cost efficient

Table 3.5: Computational Efficiency of BLARS with Non-additive Cost Using Smaller γ Values to Save Search Time for Larger Ones.

		Number of Times	Proportions of Full Search
λ	γ	to Call *lars*	(Times/1023)
20	0.5	16	1.56%
	1	17	1.66%
	7	24	2.35%
	15	27	2.64%
	35	9	0.88%
200	0.5	9	0.88%
	1	13	1.27%
	7	19	1.86%
	15	18	1.76%
	35	9	0.88%

variable selection process. Table 3.6 displays the variable selection results for some combinations of λ and γ. Corresponding to Table 3.6, the effect of γ on the error sum of squares (SSE) and cost are shown in Table 3.7. For fixed λ and increasing γ, at first the percentage of SSE in total loss is decreasing, but as the percentage of cost increases, fewer and cheaper variables are selected and the percentage of SSE in the total losses increases again.

Table 3.8 and Table 3.9 show the computational efficiency of BLARS dealing with non-additive cost for the 64-term design matrix with $\lambda = 60$. Since there are 64 potential predictors, we need to consider $2^{64} - 1$ potential models. Running *lars* on all of those for a full search is clearly infeasible. When using the BLARS approach, the number of times the *lars* is called is reduced dramatically.

Table 3.6: Coefficients of Full Models Resulting from Cost-efficient Variable Selection.

		Estimated Coefficients									
λ	γ	AGE	SEX	BMI	BP	S3	S5	S6	AGE^2	BMI^2	S6^2
60	0	0	-11.35	5.40	0.88	-0.71	42.71	0.086	0.003	0.068	0.017
	0.5	0	-11.35	5.40	0.88	-0.71	42.71	0.086	0.003	0.068	0.017
	1	0	-11.42	5.42	0.88	-0.73	43.66	0	0.002	0.069	0
	7	0	-4.95	6.00	0.81	0	48.68	0	0.003	0.065	0
	15	0.054	-1.86	8.22	1.20	0	0	0	0.0007	0	0
	35	0.350	0	9.63	0	0	0	0	0.0061	0	0

λ	γ	AGE:SEX	AGE:BMI	SEX:BMI	AGE:BP	SEX:BP	BMI:BP	AGE:S5	AGE:S6
60	0	0.82	0	0	0.008	0.019	0.064	0.070	0.004
	0.5	0.82	0	0	0.008	0.019	0.064	0.070	0.004
	1	0.79	0	0	0.010	0.070	0.072	0.154	0
	7	0.88	0	0	0.012	0.058	0.067	0.152	0
	15	0.53	0	0.58	0.028	0.083	0.050	0	0
	35	0.52	0.033	1.62	0	0	0	0	0

Table 3.7: Effect of γ on SSE and Non-additive Cost for Full Models.

λ	γ	SSE (thousand)	Percentage of SSE in total loss	Cost ($n\gamma C$) (thousand)	Percentage of Cost in total loss
60	0	1221	91.33%	0	0%
	0.5	1221	89.48%	28	2.02%
	1	1234	88.87%	42	3.02%
	7	1281	80.83%	201	12.69%
	15	1520	89.15%	99	5.83%
	35	1663	91.76%	77	4.27%

3.3.3 Optimal Values Using Non-additive Costs

We choose the optimal tuning parameter and optimal model based on the criterion of BIC for the lasso (Equation 2.2.2) proposed by Zou *et al.* (2007). The optimal value of λ based on BIC depends on the value of γ. For the 10-variable design matrix, the optimal λ value is 19.98 for lasso selection ($\gamma = 0$) based on BIC. For $\gamma = 1$, the optimal λ was found to be 9.21. Compared with the lasso solution ($\gamma = 0$), the optimal model using BLARS with $\gamma = 1$ contains two less predictors: S1 and S6, which is due to the cost effect, and the total cost decreased by 38.7%. This "cheaper" model is selected by sacrificing a 1.0% increase of SSE, and 5.6% decrease in $\sum_{j=1}^{p} |\beta_j|$. Table 3.10 displays the optimal lasso model and the optimal BLARS model with several γ values.

For the 64-variable design matrix and $\gamma = 1$, the optimal λ was found to be $\lambda = 72.25$. The optimal model contains 10 variables. We display the optimal models using BLARS and the optimal lasso model ($\gamma = 0$) based on BIC criterion in Table 3.11. Compared with the lasso solution, the total cost using BLARS with $\gamma = 1$

Table 3.8: Computational Efficiency of BLARS for Full Model with Non-additive Cost, for Fixed Choices of λ and γ.

		Number of Times	Proportions of Full Search
λ	γ	to Call *lars*	Times/$(2^{64}-1)$
60	0.5	39	10^{-18}
	1	76	10^{-18}
	7	96	10^{-17}
	15	132	10^{-17}
	35	319	10^{-17}

Table 3.9: Computational Efficiency of BLARS for Full Model with Non-additive Cost Using Smaller γ Values to Save Search Time for Larger Ones.

		Number of Times	Proportions of Full Search
λ	γ	to Call *lars*	Times/$(2^{64}-1)$
60	0.5	39	10^{-18}
	1	76	10^{-18}
	7	66	10^{-18}
	15	64	10^{-18}
	35	58	10^{-18}

Table 3.10: Optimal Main-effect Models Chosen by Lasso and BLARS with Non-additive Cost.

		Estimated Coefficients									
Method	γ	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6
Lasso	0	0	-18.85	5.63	1.02	-0.14	0	-0.82	0	46.92	0.23
BLARS	1	0	-20.88	5.63	1.09	0	0	-1.02	0	43.01	0
	7	0	-10.00	6.33	0.94	0	0	0	0	49.17	0
	15	0	0	7.91	1.19	0	0	0	0	0	0
	35	0.40	0	9.71	0	0	0	0	0	0	0

decreased by 24.0%, by excluding the cost of *S6* in the terms *S6*^{2} and *AGE:S6*. The SSE also decreased by 1.0%, but $\sum_{j=1}^{p} |\beta_j|$ increased by 6.1%.

Table 3.11: Optimal Full Models Chosen by Lasso and BLARS with Non-additive Cost.

		Estimated Coefficients								
Method	γ	AGE	SEX	BMI	BP	S3	S5	AGE^2	BMI^2	S6^2
Lasso	0	0	-6.03	5.43	0.78	-0.57	42.09	0	0.045	0.013
BLARS	1	0	-9.36	5.41	0.83	-0.68	43.17	0	0.060	0
	7	0	0	5.83	0.68	0	45.71	0	0.034	0
	15	0	0	7.83	1.11	0	0	0	0	0
	35	0.47	0	9.99	0	0	0	0.012	0	0
Method	γ	AGE:SEX	AGE:BMI	SEX:BMI	AGE:BP	BMI:BP	AGE:S5	AGE:S6		
Lasso	0	0.60	0	0	0.0064	0.051	0	0.0039		
BLARS	1	0.73	0	0	0.0096	0.067	0.112	0		
	7	0.58	0	0	0.0088	0.050	0	0		
	15	0.30	0	0	0.0190	0.033	0	0		
	35	0.67	0.055	2.17	0	0	0	0		

3.4 Conclusion and Discussion

BLARS will be usable whenever costs are a non-decreasing function of the set of selected variables. That includes both additive and non-additive cost structures. This chapter improved the BLARS method so that it is not only more efficient but also suitable for dealing with non-additive cost. The example displayed the cost-efficient variable selection results for both a main effect model and a full model.

To make our BLARS method more practical, it is helpful to build a software implementation. With an R package, the cost-efficient variable selection process will become a routinely available procedure for health researchers once they have the cost information on collecting the data. We name the R package *branchLars*, and describe it in Chapter 4.

Chapter 4

SOFTWARE IMPLEMENTATION

4.1 Introduction to the R package branchLars

The BLARS methods dealing with both additive cost and non-additive cost have been developed in the previous chapters. Having a software implementation for BLARS will make it more practical, and the health researchers can routinely apply the cost-efficient variable selection strategy in statistical model building provided that they have the information on the costs of collecting the data. We built the R package *branchLars* which implements the BLARS algorithm. It will be described in detail in this chapter.

The R package *branchLars* depends on the R packages *lars* and *Rgraphviz*. Since the *lars* function is called in the BLARS process, the R package *lars* should be installed in advance. We also want to display the BLARS search tree visually for a pre-fixed pair of λ and γ to give the user an imaginable idea of the BLARS searching process, and the R package *Rgraphviz* helps to realize this object. We presume the R package *Rgraphviz* has also been installed in advance.

The rest of this chapter is organized as follows. Section 4.2 describes the functions in the R package *branchLars*. The usage of the functions in cost-efficient variable selection is illustrated using an example in Section 4.3. Section 4.4 contains conclusion and discussion.

4.2 Package Description

The main function of the package is *branchLars* which implements the BLARS search method. The usage of the function is as follows.

```
branchLars(X, y, lambda, cost, gamma=1, sumC=0)
```

The $n \times p$ matrix X contains p potential predictors $\mathbf{x}_1, \ldots, \mathbf{x}_p$, each of them has to be standardized to have mean 0 and norm 1. The response vector $\mathbf{y}$ has to be standardized to have mean 0. The function *standardize* in the package takes $X, \mathbf{y}$ in the original scale as inputs, standardizes the response $\mathbf{y}$ and the predictors in X, and outputs the standardized $X, \mathbf{y}$ and a vector of the norm values for the original predictors. Another function *standardizeG* does the similar job which returns a matrix X also containing standardized squared terms and two-way interaction terms. Note that the estimated coefficients returned by the function *branchLars* are in the transformed scale, and the function *unstandardize* could be used to transform the coefficients back to the original scale.

A fast effective way of selecting the tuning parameter λ is important in practice. We use BIC for the lasso (Equation 2.2.2) as the default tuning parameter and model selection criteria in the *branchLars* package for its simplicity and effectiveness. The parameter γ is a user-defined weight imposed on costs, reflecting the level of reluctance to use high cost variables. When $\gamma = 0$, we ignore the costs and the selection becomes the standard lasso variable selection. The higher the γ value, the more reluctance the user to select high cost variables. Thus when the user inputs a higher value to γ, the *branchLars* function will be less likely to select higher cost variables. There is a trade off between an expensive but more accurate model and a cheap model with lower accuracy. The assignment of a γ value is thus based to a large extent on the opinions and judgments of the user or the decision maker. Sometimes the user has to use a higher γ because of budget constraint. Once the user chooses a value for γ, the optimal value of λ can be found by the function *lambdaOpt* based on the default

BIC criteria, and the corresponding optimal statistical model could then be selected by calling the *branchLars* function again with the optimal λ as the input. The user could use their own preferred method, such as *Cp* for the lasso (Efron *et al.*, 2004) or cross validation, to select an optimal tuning parameter, then call the *branchLars* function to build the cost-efficient statistical model.

Given a fixed value of λ in BLARS search process, the value of $C(\alpha_1 c_1, \ldots, \alpha_p c_p)$ in the objective function 3.2.1 of the optimal model can not increase when we increase the value of γ. Thus when we increase γ from γ_1 to γ_2, we could use the value of $C(\alpha_1 c_1, \ldots, \alpha_p c_p)$ returned by *branchLars* using γ_1 as the input value of *sumC* in the call for *branchLars* when using γ_2, so that the *branchLars* can prune a branch if $C(\alpha_1 c_1, \ldots, \alpha_p c_p)$ of this branch is larger than the *sumC* value. We could also ignore all those variables that have higher cost than the sum of the costs of the variables selected by *branchLars* using γ_1 to reduce the searching time.

The parameter *cost* in the *branchLars* function is a two-element list. The first element is a vector containing all the group costs and additional individual costs for each variable. The second element is also a list with length p, where p equals the number of potential predictors. These p terms have the same names as the p potential predictors. The i^{th} term contains the pointers to the cost values which could be used to calculate the cost for the i^{th} predictor, where $i = 1, \ldots, p$. For the simple additive costs, suppose we have three variables x_1, x_2, x_3 with the costs c_1, c_2, c_3 respectively, we create the parameter *cost* using the following R code:

```
cost <- list()
cost[[1]] <- c(c1, c2, c3)              # the first element
cost[[2]] <- list(1, 2, 3)              # the second element
names(cost[[2]]) <- c("x1", "x2", "x3")
```

The *cost* should then be displayed as

```
cost
[[1]]
 [1]  c1  c2  c3
```

```
[[2]]
[[2]]$x1
[1] 1

[[2]]$x2
[1] 2

[[2]]$x3
[1] 3
```

where the first term of the second element has the name x_1 and contains the pointer 1, which points to the first cost value c_1. For the non-additive costs as in Figure 3.2, the *cost* could be creased using the following R code:

```
cost <- list()
cost[[1]] <- c(c1, c2, c3, c5)                       # the first element
cost[[2]] <- list(1, 2, c(1,3), 1, c(2,4))           # the second element
names(cost[[2]]) <- c("x1", "x2", "x3", "x4", "x5")
```

Then the *cost* should be displayed as:

```
cost
[[1]]
 [1]  C1  C2  C3  C5

[[2]]
[[2]]$x1
[1] 1

[[2]]$x2
[1] 2

[[2]]$x3
[1] 1 3

[[2]]$x4
[1] 1

[[2]]$x5
[1] 2 4
```

where the last term of the second element has the name x_5 and contains the pointers 2 and 4, which point to the second and fourth cost values C_2 and C_5. The cost of variable x_5 is then $C_2 + C_5$ given variable x_2 has not been selected. If we want to include the squared and two-way interaction terms, such as x_1^2 and $x_1 : x_3$, we need to include the pointer 1 for the term x_1^2 and pointers $(1, 3)$ for the term $x_1 : x_3$ in the second element of *cost*. All these additional pointers for squared and two-way interaction terms could be added in the cost variabale by calling the function *standardizeG*.

With the above cost structure, the function *branchLars* will automatically update the costs of each undetermined variables based on which variables have been selected into the model. The function *branchLars* calls a build-in function *buildCostFun*, which takes the parameter *cost* as the input and outputs a function *costFun*. This function *costFun* is used by *branchLars* to calculate the sum of the costs of the selected variables (a scalar and the first element of the *costFun* outputs) and update the costs of the undetermined varilables (a vector and the second element of the *costFun* outputs) at each BLARS step. A vector *alpha* of length p is the only input of the function *costFun*, and it contains the values of $-1, 0$ and 1, where 1 means the corresponding variable has been selected, 0 means the variable has been excluded, and -1 means the variable is undetermined. This vector *alpha* is also updated by the function *branchLars* at each step. The users could input the parameter *cost* in their own specific structure, provided that they can write their corresponding function *buildCostFun*.

The function *branchLars* returns a BLARS object. The estimated regression coefficients in the transformed scale could be displayed using function *print(BLARS object)*, and the coefficients in the original scale could be obtained using the function *unstandardize(BLARS object, norm values)*. The function *summary(BLARS object)* shows the optimized objective value with its three components (SSE, lasso type l_1-penalty, and total cost) corresponding to the input pair of λ and γ. The function *predict(BLARS object, newdata)* could be used to predict the responses for a new dataset X'. We could find the times that the function *lars* have been called in the

BLARS search process by accessing the *evals* component of the BLARS object. The BLARS search tree could be visualized using the function *drawTree(BLARS object)*. The function *bic(BLARS object)* and *cp(BLARS object)*return the BIC and Cp values of the returned BLARS model.

Table 4.1 summarizes the functions available in the R package *branchLars*.

branchLars	Input	Output	Description
branchLars	$\boldsymbol{X}$, $\boldsymbol{y}$, *lambda*, *cost*, *gamma*, *sumC*	branchLars object	Cost-efficient variable selection and estimation for given λ and γ
standardize	$\boldsymbol{X}$, $\boldsymbol{y}$	$\boldsymbol{X}s$, $\boldsymbol{y}s$, *normX*	Standardize the design matrix $\boldsymbol{X}$ and response $\boldsymbol{y}$
standardizeG	$\boldsymbol{X}s$, *normX*, *cost*	$\boldsymbol{X}s$, *normX*, *cost*	Add standardized squared and two-way interaction terms to design matrix; return corresponding cost variable
unstandardize	BLARS object, *normX*	regression coefficients	Transform the regression coefficients back to the original scale
print	BLARS object	regression coefficients	Print the estimated regression coefficients in transformed scale
summary	BLARS object	optimal total loss	Summarize the optimal total loss and its three components
predict	BLARS object, new data	predicted responses	Make prediction
drawTree	BLARS object	the search tree	Visualize the search process
bic	BLARS object	BIC value	Calculate the BIC value for a BLARS model
cp	BLARS object	Cp value	Calculate the Cp value for a BLARS model
lambdaOpt	$\boldsymbol{X}$, $\boldsymbol{y}$, *cost*, *gamma*, *lower*, *upper*, *method*, *tol*	Optimal λ, BIC/Cp, *sumC*	Search the optimal λ from a search range for a given γ

Table 4.1: Functions in the *branchLars* Package

4.3 Example of Use of the Package

We apply the functions in package *branchLars* to the diabetes data used by Efron *et al.* (2004). The data consists of 10 variables: age, sex, body mass index (BMI), average blood pressure (BP), and S1 to S6 representing 6 serum measurements. For the purpose of illustration, we assign the non-additive costs to the 10 variables as shown in Table 3.1.

4.3.1 Cost Efficient Variable Selection for Fixed Value of λ

Let X be the design matrix and $\mathbf{y}$ the response vector. We first standardize the data so that the covariates have mean 0 and unit length, and the response has mean 0. Then we construct the parameter *cost*.

```
> library(branchLars)

Loading required package: lars
Loading required package: Rgraphviz
Loading required package: graph
Loading required package: grid

> data(diabetes)
> X <- as.matrix(diabetes[, 1:10])
> y <- diabetes$Y
> Xy <- standardize(X, y)
> Xs <- Xy$X       # standardized design matrix
> ys <- Xy$y       # standardized response vector
> normx <- Xy$normX
> costs <- list()
> costs[[1]] <- c(0, 5, 10, 20, rep(30, 6))
> costs[[2]] <- list(1, 1, 2, 3, c(4,5), c(4,6),
                     c(4,7),c(4,8), c(4,9), c(4,10))
> names(costs[[2]]) <- colnames(Xs)
> costs

[[1]]
 [1]  0  5 10 20 30 30 30 30 30 30
```

```
[[2]]
[[2]]$AGE
[1] 1

[[2]]$SEX
[1] 1

[[2]]$BMI
[1] 2

[[2]]$BP
[1] 3

[[2]]$S1
[1] 4 5

[[2]]$S2
[1] 4 6

[[2]]$S3
[1] 4 7

[[2]]$S4
[1] 4 8

[[2]]$S5
[1] 4 9

[[2]]$S6
[1] 4 10
```

We choose the tuning parameter $\lambda = 30$ as an example to illustrate the cost efficient variable selection process. The parameter γ is a user-defined weight imposed on costs, reflecting the level of reluctance to use high cost variables. Suppose the user provides a value of $\gamma = 0.8$, then we can select and estimate the predictors using function *branchLars*. Function *summary* gives the total loss and its three components for the selected regression model; function *unstandardize* transforms the regression coefficients back to the original scale.

```
> result1 <- branchLars(Xs, ys, lambda=30, costs, gamma=0.8)
> print(result1)

Call:
branchLars(X = Xs, y = ys, lambda = 30, cost = costs, gamma = 0.8)

Regression Coefficient:
     AGE       SEX      BMI       BP S1 S2        S3 S4       S5 S6
[1,]   0 -181.4176 519.4402 295.1167  0  0 -248.7673  0 466.3076  0

> summary(result1)

Call:
branchLars(X = Xs, y = ys, lambda = 30, cost = costs, gamma = 0.8)

Optimal Total Loss and its Components:
Total_Loss        RSS L1_penalty       Cost
1376942.45 1292018.97   51331.48   33592.00

> unstandardize(result1, normx)

     AGE       SEX      BMI       BP S1 S2        S3 S4       S5 S6
[1,]   0 -17.29304 5.598589 1.016043  0  0 -0.915871  0 42.50675  0

> drawTree(result1)
```

The BLARS search tree can be visualized using the function *drawTree*. Figure 4.1 shows the search tree for $\lambda = 30$ and $\gamma = 0.8$, where 1 means the variable is selected and the red path is the optimal search path.

There are 10 potential predictors, hence $2^{10} = 1024$ different $\boldsymbol{\alpha}$ values. If no variable is selected with an initial *lars* call for some λ values, the search process stops there. For general situations, we usually need to call the *lars* function up to 1023 times to make a full search. Note that there is no need to call *lars* when all $\alpha_j = 0$ which means no variable is in the model. When using the BLARS approach, the number of times the *lars* is called is much reduced. The *branchLars* object returned by function *branchLars* contains an element *evals* showing the total times to call *lars* function.

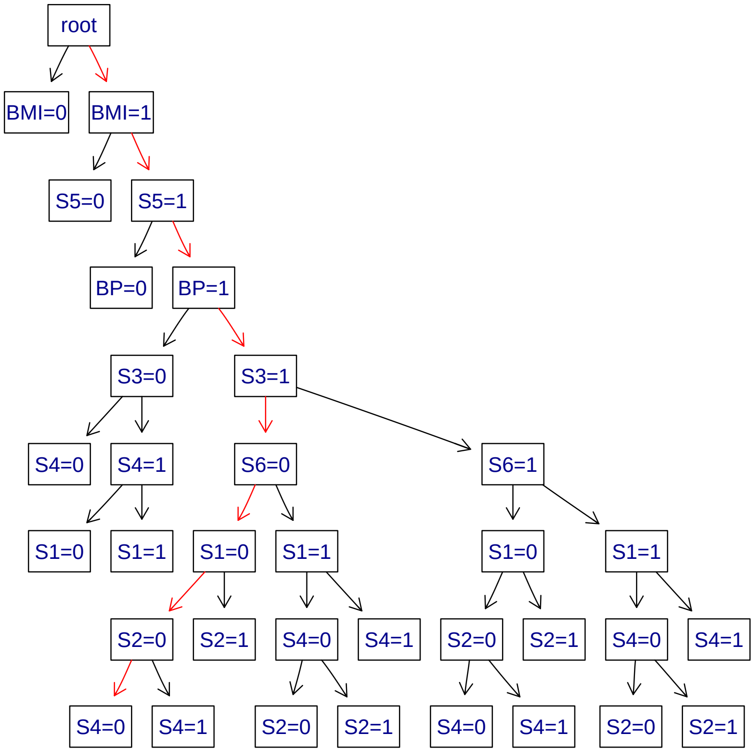

Figure 4.1: Search Tree for $\lambda = 30$ and $\gamma = 0.8$ with 10-Variable Design Matrix

With fixed λ, we could take the sum of costs of previously selected variables using a lower γ as the input value of *sumC* to reduce the computing time. For example, the times to call *lars* function is 38 when $\lambda = 30$ and $\gamma = 30$. If we input the previous *sumC* value returned by function *branchLars* with $\gamma = 15$ into the new *branchLars* call with $\gamma = 30$, the times to call *lars* function is reduced to 9. Another time-saving approach is to ignore some variables that have higher cost than the sum of the costs of the variables selected previously using a lower γ. For instance, when $\lambda = 30$ and $\gamma = 15$, the process selects the first four variables and the sum of the costs for each patient is 15; if we increase γ, we could delete the variables S1 to S6 from the design matrix since the cost of each S1 to S6 is greater than 15, and we still get the same results, but the times to call *lars* function drops dramatically (only 3 times compared with the previous 38 when $\gamma = 30$).

```
> result2 <- branchLars(Xs, ys, lambda=30, costs, gamma=15)
> unstandardize(result2, normx)

            AGE       SEX     BMI       BP S1 S2 S3 S4 S5 S6
[1,] 0.04193434 -6.163864 8.27366 1.352763  0  0  0  0  0  0

> result3 <- branchLars(Xs, ys, lambda=30, costs, gamma=30)
> result3$evals

[1] 38

> unstandardize(result3, normx)

           AGE SEX     BMI BP S1 S2 S3 S4 S5 S6
[1,] 0.3912211   0 9.69494  0  0  0  0  0  0  0

> result4 <- branchLars(Xs, ys, lambda=30, costs, gamma=30,
+                       sumC=result2$sumC)
> result4$evals

[1] 9

> unstandardize(result4, normx)
```

```
           AGE SEX     BMI BP S1 S2 S3 S4 S5 S6
[1,] 0.3912211   0 9.69494  0  0  0  0  0  0  0

> result5 <- branchLars(Xs[,1:4], ys, lambda=30, costs[1:4], gamma=30)
> result5$evals

[1] 3

> unstandardize(result5, normx[1:4])

           AGE SEX     BMI BP
[1,] 0.3912211   0 9.69494  0
```

We could include squared terms and two-way interaction terms in the design matrix using the function *standardizeG*. Excluding the SEX^2 term, we have 64 potential predictors. The following are some results for this bigger design matrix when we assign $\lambda = 65$ and $\gamma = 1.5$.

```
> XyG <- standardizeG(Xs, normx, costs)
> XsG <- XyG$Xs[, -12]      # exclude the squared SEX term
> normX <- XyG$normX[-12]
> costsG <- XyG$cost
> costsG[[2]] <- costsG[[2]][-12]
> result6 <- branchLars(XsG, ys, lambda=65, costsG, gamma=1.5)
> result6$evals

[1] 76

> summary(result6)

Call:
branchLars(X = XsG, y = ys, lambda = 65, cost = costsG, gamma = 1.5)

Optimal Total Loss and its Components:
Total_Loss        RSS L1_penalty       Cost
 1421713.0  1239487.8   119240.2    62985.0

> unstandardize(result6, normX)
```

```
     AGE       SEX      BMI        BP S1 S2         S3 S4       S5 S6
[1,]   0 -10.57813 5.414105 0.8624607  0  0 -0.7098747  0 43.45239  0
            AGE^2      BMI^2 BP^2 S1^2 S2^2 S3^2 S4^2 S5^2 S6^2
[1,] 0.0006874186 0.06546686    0    0    0    0    0    0    0
       AGE:SEX AGE:BMI SEX:BMI      AGE:BP     SEX:BP     BMI:BP
[1,] 0.7677764       0       0 0.009862242 0.04133636 0.06988001
     AGE:S1 SEX:S1 BMI:S1 BP:S1 AGE:S2 SEX:S2 BMI:S2 BP:S2 S1:S2
[1,]      0      0      0     0      0      0      0     0     0
     AGE:S3 SEX:S3 BMI:S3 BP:S3 S1:S3 S2:S3 AGE:S4 SEX:S4 BMI:S4
[1,]      0      0      0     0     0     0      0      0      0
     BP:S4 S1:S4 S2:S4 S3:S4    AGE:S5 SEX:S5 BMI:S5 BP:S5 S1:S5
[1,]     0     0     0     0 0.1391305      0      0     0     0
     S2:S5 S3:S5 S4:S5 AGE:S6 SEX:S6 BMI:S6 BP:S6 S1:S6 S2:S6 S3:S6
[1,]     0     0     0      0      0      0     0     0     0     0
     S4:S6 S5:S6
[1,]     0     0
```

4.3.2 Optimal Models

We choose the optimal tuning parameter and optimal model based on the default BIC criterion. Function *bic* gives the BIC value for a selected BLARS model. The optimal value of λ based on BIC depends on the value of γ. For the 10-variable design matrix, the optimal λ value is 19.98 for standard lasso selection based on BIC. For $\gamma = 1$, the optimal λ was found using function *lambdaOpt* to be $\lambda = 9.213$. We could compare the optimal model when $\gamma = 1$ with the optimal model when $\gamma = 0$ (the lasso solution based on BIC criterion) in Table 3.10. The optimal model using BLARS contains two less predictors: S1 and S6, and the total cost decreased by 38.7%. This "cheaper" model is selected by sacrificing a 1.0% increase of SSE, and 5.6% decrease in $\sum_{j=1}^{p} |\beta_j|$.

```
> lassoOpt <- lambdaOpt(Xs, ys, costs, gamma=0)  # Lasso
> lassoOpt

$Optimal_Lambda
[1] 19.98117
```

```
$BIC
[1] 1.094136

$sumC
[1] 155

> modelOptLasso <- branchLars(Xs, ys, lambda=lassoOpt$Optimal_Lambda,
+                            costs, gamma=0, sumC=lassoOpt$sumC)
> modelOptLasso$evals

[1] 1

> summary(modelOptLasso)

Call:
branchLars(X = Xs, y = ys, lambda = lassoOpt$Optimal_Lambda,
    cost = costs, gamma = 0, sumC = lassoOpt$sumC)

Optimal Total Loss and its Components:
Total_Loss        RSS L1_penalty       Cost
1313612.34 1275357.12   38255.23       0.00

> unstandardize(modelOptLasso, normx)

     AGE       SEX     BMI       BP         S1 S2         S3 S4
[1,]   0 -18.85021 5.62909 1.023057 -0.1430241  0 -0.8244074  0
           S5        S6
[1,] 46.92238 0.2268591

> opt1 <- lambdaOpt(Xs, ys, costs, gamma=1)  # BLARS
> opt1

$Optimal_Lambda
[1] 9.212472

$BIC
[1] 1.076536

$sumC
[1] 95
```

```
> modelOpt1 <- branchLars(Xs, ys, lambda=opt1$Optimal_Lambda,
+                          costs, gamma=1, sumC=opt1$sumC)
> modelOpt1$evals

[1] 14

> summary(modelOpt1)

Call:
branchLars(X = Xs, y = ys, lambda = opt1$Optimal_Lambda, cost = costs,
    gamma = 1, sumC = opt1$sumC)

Optimal Total Loss and its Components:
Total_Loss        RSS L1_penalty       Cost
1346904.80 1288271.35   16643.45   41990.00

> unstandardize(modelOpt1, normx)

     AGE       SEX      BMI       BP S1 S2        S3 S4       S5 S6
[1,]   0 -20.88318 5.629415 1.090270  0  0 -1.018800  0 43.01096  0
```

For the 64-variable design matrix and $\gamma = 1$, the optimal λ was found to be $\lambda = 72.25$. The optimal BLARS model contains 10 variables, while the optimal lasso model selects 11 variables. Comparing the optimal model when $\gamma = 1$ with the optimal model when $\gamma = 0$ (the lasso solution) based on BIC criterion, we found that the total cost using BLARS decreased by 24.0%, by excluding the cost of S6 in the terms S6^2 and AGE:S6. The SSE also decreased by 1.0%, but $\sum_{j=1}^{p} |\beta_j|$ increased by 6.1%.

```
> lassoOpt2 <- lambdaOpt(XsG, ys, costsG, gamma=0)  # Lasso
> lassoOpt2

$Optimal_Lambda
[1] 91.47057

$BIC
[1] 1.171799

$sumC
[1] 125
```

```
> modelOptLasso2 <- branchLars(XsG, ys, lambda=lassoOpt2$Optimal_Lambda,
+                          costsG, gamma=0, sumC=lassoOpt2$sumC)
> modelOptLasso2$evals

[1] 1

> summary(modelOptLasso2)

Call:
branchLars(X = XsG, y = ys, lambda = lassoOpt2$Optimal_Lambda,
    cost = costsG, gamma = 0, sumC = lassoOpt2$sumC)

Optimal Total Loss and its Components:
Total_Loss        RSS L1_penalty       Cost
 1413525.8  1260439.0   153086.8        0.0

> unstandardize(modelOptLasso2, normX)

     AGE       SEX      BMI        BP S1 S2         S3 S4       S5 S6
[1,]   0 -6.025656 5.426364 0.7800494  0  0 -0.5704826  0 42.08848  0
     AGE^2      BMI^2 BP^2 S1^2 S2^2 S3^2 S4^2 S5^2       S6^2
[1,]     0 0.04450851    0    0    0    0    0    0 0.01277153
       AGE:SEX AGE:BMI SEX:BMI     AGE:BP SEX:BP     BMI:BP AGE:S1
[1,] 0.5955218       0       0 0.00637261      0 0.05141858      0
     SEX:S1 BMI:S1 BP:S1 AGE:S2 SEX:S2 BMI:S2 BP:S2 S1:S2 AGE:S3
[1,]      0      0     0      0      0      0     0     0      0
     SEX:S3 BMI:S3 BP:S3 S1:S3 S2:S3 AGE:S4 SEX:S4 BMI:S4 BP:S4 S1:S4
[1,]      0      0     0     0     0      0      0      0     0     0
     S2:S4 S3:S4 AGE:S5 SEX:S5 BMI:S5 BP:S5 S1:S5 S2:S5 S3:S5 S4:S5
[1,]     0     0      0      0      0     0     0     0     0     0
          AGE:S6 SEX:S6 BMI:S6 BP:S6 S1:S6 S2:S6 S3:S6 S4:S6 S5:S6
[1,] 0.003878718      0      0     0     0     0     0     0     0

> opt2 <- lambdaOpt(XsG, ys, costsG, gamma=1, lower=70, upper=90) # BLARS
> opt2

$Optimal_Lambda
[1] 72.24969

$BIC
[1] 1.147681

$sumC
[1] 95
```

```
> modelOpt2 <- branchLars(XsG, ys, lambda=opt2$Optimal_Lambda,
+                          costsG, gamma=1, sumC=opt2$sumC)
> modelOpt2$evals

[1] 44

> summary(modelOpt2)

Call:
branchLars(X = XsG, y = ys, lambda = opt2$Optimal_Lambda, cost = costsG,
    gamma = 1, sumC = opt2$sumC)

Optimal Total Loss and its Components:
Total_Loss        RSS L1_penalty       Cost
 1417805.3  1247493.2   128322.1    41990.0

> unstandardize(modelOpt2, normX)

     AGE       SEX      BMI        BP S1 S2        S3 S4       S5 S6
[1,]   0 -9.364981 5.412267 0.8345961  0  0 -0.676748  0 43.17241  0
     AGE^2     BMI^2 BP^2 S1^2 S2^2 S3^2 S4^2 S5^2 S6^2   AGE:SEX
[1,]     0 0.0604145    0    0    0    0    0    0    0 0.7298097
     AGE:BMI SEX:BMI      AGE:BP SEX:BP     BMI:BP AGE:S1 SEX:S1
[1,]       0       0 0.009575883      0 0.06716839      0      0
     BMI:S1 BP:S1 AGE:S2 SEX:S2 BMI:S2 BP:S2 S1:S2 AGE:S3 SEX:S3
[1,]      0     0      0      0      0     0     0      0      0
     BMI:S3 BP:S3 S1:S3 S2:S3 AGE:S4 SEX:S4 BMI:S4 BP:S4 S1:S4 S2:S4
[1,]      0     0     0     0      0      0      0     0     0     0
     S3:S4    AGE:S5 SEX:S5 BMI:S5 BP:S5 S1:S5 S2:S5 S3:S5 S4:S5
[1,]     0 0.1121557      0      0     0     0     0     0     0
     AGE:S6 SEX:S6 BMI:S6 BP:S6 S1:S6 S2:S6 S3:S6 S4:S6 S5:S6
[1,]      0      0      0     0     0     0     0     0     0
```

4.4 Conclusion and Discussion

The main function *branchLars* in the R package *branchLars* implements the BLARS algorithm developed in the previous 2 chapters. Other functions in the package facilitate the main function with respect to either the input values or the outputs. A detailed example was used in this chapter to illustrate the usage of the functions in the package in a cost-efficient variable selection data analysis process.

The optimal value of λ and the corresponding optimal branching LARS model are selected based on the default model selection criteria, BIC for the lasso (Zou *et al.*, 2007), in the *branchLars* package. Researchers may have their preferred model selection criteria other than BIC. The function *lambdaOpt* can be adapted based on other selection criteria to find the optimal λ value. The corresponding optimal BLARS model can still be obtained using the function *branchLars* once the optimal λ has been selected.

Chapter 5

REAL DATA APPLICATION - ACT PROJECT

5.1 Description of the Project

A study was conducted in Southwestern Ontario to assess factors which may influence the outcomes of clients with severe mental illness (SMI) receiving care from the Assertive Community Treatment (ACT)(Lehman and Steinwachs, 1998) service. A total of 233 SMI patients who meet the Ontario Ministry of Health standards for ACT (Ministry of Health, 1998) and who are either already receiving ACT services or are entering any ACT teams in Southwestern Ontario (London, Windsor, Sarnia, Waterloo, Milton, etc.) were recruited. These patients were diagnosed as having psychosis or multiple co-morbid psychiatric and physical disorders, having a history of high hospital use, long-term illness, high needs, and low functioning.

There are about 19 potential predictive factors. Some of them are population characteristic variables, such as Age, Sex, Marital Status, Diagnosis, Duration of illness, and 3 Colorado Client Assessment Record (CCAR) (Ellis *et al.*, 1984) subscales revised for use in Southwest Ontario (Ministry of Health, 1999) (level of function subscale, substance use subscale, and employment status subscale). The other variables measure the treatment, rehabilitation and support services actually delivered, such as number of months in ACT, medications prescribed, and number of contacts by ACT staff per month. The implementation system is measured by the fidelity of team to the ACT model using Dartmouth ACT Scale (Teague *et al.*, 1995, 1998). Variables that mediate the treatment effects (intervening variables that occur after clients have been assigned to treatment but before measurement of longer term client outcomes) include Working Alliance Inventory, Empowerment Scale, Drug Attitude Inventory, Adherence to Medication Scale, and Present State Exam-insight Score. Table 5.1 presents the names and descriptions of the variables used in the analysis of the ACT project. Long term outcome is the Overall CCAR Score, which is the overall degree of problem severity (a larger score associates with a higher level of problem severity) and was measured at 12 and 24 months after enrollment in the project.

Our role in this study is to assess what cost-efficient factors influence outcomes of clients with SMI receiving care from ACT. We want to find the risk factors not only with higher prediction accuracy but also cheaper and easier to collect the data so that we can reduce the burden of ACT team, patients, and health care system.

5.2 Why Selecting Cost-efficient Factors

Since the sources of data collection are different, the cost of collecting data is different for the potential predictors. In the ACT project, data were collected from the following sources.

- Client self-reports: research assistants meet with clients to administer collecting the data including Working Alliance Inventory, Present State Exam-insight score, Empowerment Scale, and Drug Attitude Inventory.

- ACT clinicians: research assistants visit the ACT team office and let the clinicians who know the clients best to give the scores, such as Adherence to Medication Scale, and CCAR subscales.

- Client records: research assistants visit the ACT team offices and extract the data from charts about the clients, for example, Age, Marital status, Diagnosis, Duration of illness, Medications prescribed and Total service use.

Table 5.1: Potential Predictive Factors in ACT Project.

Predictors	Description
Age	Age in years
Sex	1: Female ; 0: Male
Mstatus	Marital Status, 1: Married or Common-law; 0: Otherwise
CoMorbid	Number of co-morbid diagnoses
Duration	Number of years since first diagnosis
Lifetime	Lifetime days in hospital
Jail	Ever in jail, 1: No; 0: Yes
EmpSC	CCAR employment subscale
	1: Employed (full-time or part-time); 0: Otherwise
SubSC	CCAR substance use subscale
	A larger score associates with a higher level of substance abuse
FunSC	CCAR functioning subscale
	A larger score associates with a lower level of functioning
ACTmonth	Total service use: number of months in ACT
Medtype	Medications prescribed: number of medication categories
Contacts	Intensity of contacts: average number of contacts per month by ACT staff
DACTS	Fidelity of team to ACT model: Dartmouth ACT Scale
	A larger score associates with a higher level of fidelity to ACT model

Predictors	Description
WAIscore	Therapeutic alliance: Working Alliance Inventory A larger score associates with a higher level of partnership or alliance between patient and therapist
PSEscore	Insight into psychosis: Present State Exam-insight score A larger score associates with a lower level of client's insight into psychotic symptoms
EMPscore	Empowerment scale A larger score associates with a higher level of client's participation in their own recovery
DAIscore	Satisfaction with medications: Drug Attitude Inventory A larger score associates with a higher level of client's satisfaction with medication
MEDCscore	Medication compliance: Adherence to medication scale A larger score associates with a lower level of client's adherence to medication

- Hospital archives: research assistants visit hospitals to extract data from hospital archive, for instance, Lifetime days in hospital.

- ACT team's staff activity records: research assistants extract data from ACT team's staff activity records, such as Intensity of Contacts.

- ACT coordinators: research assistants give the Dartmouth ACT Scale to the ACT team coordinator who fills out the required information.

The data that involves the professional work of clinicians cost higher than the data from the work of research assistants. On the other hand, the client self-reported data is harder to get than the data extracted from hospital archives due to the fact that the clients have severe mental illness. Inexpensive predictors may have the similar statistical importance as costly predictors, so inexpensive predictors could replace costly predictors by sacrificing minimal prediction accuracy while reducing the cost burden. We can build "expensive" or "cheap" models by selecting different sets of variables, and a more accurate but costly model is not necessarily better than a less accurate but cheaper model. A model is more cost-efficient than another one if it costs less but gives almost the same prediction accuracy, or it costs much less but gives only slightly less prediction power. The health researchers could make an overall judgement about the most efficient combination of variables.

5.3 Cost Structure

The cost of collecting the data has two components in the ACT project. The first is the monetary cost for human labour, time, material, equipment, compensation paid to the clients in some research activities, etc. For example, clinicians are paid for their labour to score the Adherence to Medication Scale, and the CCAR Subscales; the research assistants are paid to visit the ACT team offices and extract the data from charts about the patients, such as Duration of Illness, Number of Months in ACT.

The Duration of Illness and Number of Months in ACT cost less than Adherence to Medication Scale and the CCAR Subscales due to the professional work of clinicians. The second is the level of difficulty to get an answer or a value for a potential predictor. The clients we dealt with are the patients with severe mental illness. They may refuse to answer some questions or refuse to give information. Some results reported from the clients may need to be double checked or traced. In this sense, the client self-reported Working Alliance Inventory is much harder to get than Age, Marital Status, and Number of Months in ACT, which are simply obtained by chart extraction. These result in some variables being more "expensive" than others.

Grouping effects of cost is the next issue we need to take into account. A grouping effect occurs when one variable is selected some other variables become free or cheaper. For example, the data extracted from charts, such as Sex and Marital Status, have a total group cost for traveling to the ACT team office and the work of chart extraction by the research assistant. If Sex is selected in the model, Marital Status can be assumed free if it is also selected. Another example is the CCAR. Clinicians are trained to score the CCAR scales for their patients. The 3 CCAR subscales have a group cost for the training and some work in common, and each of them also has an additional individual cost since the clinicians have to spend a certain amount of time on each of them. If one of the CCAR subscales is selected in the model, the cost is the sum of the group cost and the additional individual cost; if another subscale is also selected, the cost for the second one becomes cheaper (only the additional individual cost).

The two components of costs of the potential predictors are estimated between 0 and 100 by the ACT project researcher and coordinator, and are listed in Table 5.2, where both monetary cost and level of difficulty consist of two parts: group cost and additional individual cost. We consider an overall cost for each predictive factor, which is a combination of the above two components. One predictor costs more than another if this predictor is more expensive overall. Since the scales of the two components are comparable (with minimum 0 and maximum 100), one simple way to combine them is to use summation. For convenience, we divide the combined costs by 200. Table 5.3 displays the combined cost information.

5.4 Data Analysis

5.4.1 Analysis without Considering Cost

Without considering cost, LARS can be directly used to provide the lasso solution. We estimate the regression coefficients to minimize the loss function

$$\hat{\boldsymbol{\beta}}_{(lasso)} = \arg\min_{\boldsymbol{\beta}} \left\{ \left\| \mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j \right\|^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\},$$

where $\lambda \geq 0$ is a regularization or tuning parameter (Efron *et al.*, 2004).

The adaptive lasso can also be solved by LARS using a transformation to the design matrix. In this case, we estimate the regression coefficients to minimize the loss function

$$\hat{\boldsymbol{\beta}}^{*(n)} = \arg\min_{\boldsymbol{\beta}} \left\{ \left\| y - \sum_{j=1}^{p} x_j \beta_j \right\|^2 + \lambda_n \sum_{j=1}^{p} \hat{w}_j |\beta_j| \right\},$$

where the known weights vector $\hat{\mathbf{w}} = (\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_p)^T$ is data-dependent. We use the form $\hat{\mathbf{w}} = |\hat{\boldsymbol{\beta}}_{(ols)}|^{-1}$ (Zou, 2006) in this thesis , where $\hat{\boldsymbol{\beta}}_{(ols)}$ is the regression coefficient estimate vector from the ordinary least squares fit.

We use both BIC for the lasso (Zou *et al.*, 2007) and *Cp* (Efron *et al.*, 2004) as the tuning parameter and model selection criteria. Based on the lasso model selected by BIC, number of months in ACT, average number of contacts per month, CCAR substance use subscale, and CCAR functioning subscale are important predictors for the response (overall CCAR score). When the number of months in ACT increases one month, the overall CCAR score decreases 0.0061 unit; if the average number of contacts increases one per month, the overall CCAR score increases 0.0028 unit;

Table 5.2: Two Components of Costs.

	Monetary Cost		Level of Difficulty	
Predictors	Group	Additional	Group	Additional
Age	15	0	10	0
Sex		0		0
Mstatus		0		0
CoMorbid		0		0
Duration		10		10
ACTmonth		0		0
Medtype		0		0
Contacts	25	0	20	0
Jail	20	0	30	0
MEDCscore	20	0	30	0
WAIscore	30	0	30	0
PSEscore		0		0
EMPscore		0		0
DAIscore		0		0
Lifetime	30	0	70	0
EmpSC	30	20	50	20
SubSC		20		20
FunSC		20		20
DACTS	60	0	100	0

Table 5.3: Overall Costs.

	Overall Cost		Re-scaled Cost	
Predictors	Group	Additional	Group	Additional
Age	25	0	0.125	0
Sex		0		0
Mstatus		0		0
CoMorbid		0		0
Duration		20		0.1
ACTmonth		0		0
Medtype		0		0
Contacts	45	0	0.225	0
Jail	50	0	0.25	0
MEDCscore	50	0	0.25	0
WAIscore	60	0	0.3	0
PSEscore		0		0
EMPscore		0		0
DAIscore		0		0
Lifetime	100	0	0.5	0
EmpSC	80	40	0.4	0.2
SubSC		40		0.2
FunSC		40		0.2
DACTS	160	0	0.8	0

the overall CCAR score will increase 0.068 and 0.16 unit if the CCAR substance use subscale and functioning subscale increase one unit respectively. Many more variables are shown in the lasso model selected by *Cp*, where ever in jail, number of months in ACT, average number of contacts per month, adherence to medication scale, working alliance inventory, empowerment scale, drug attitude inventory, CCAR employment subscale, CCAR substance use subscale, CCAR functioning subscale, and Dartmonth ACT scale are all important predictors for the overall CCAR score.

Table 5.4 displays the optimal lasso model and adaptive lasso model when we use BIC as the model selection criterion, and Table 5.5 gives the optimal lasso model and adaptive lasso model when we use *Cp* as the model selection criterion. Compared with the lasso solution, adaptive lasso gives a more parsimonious model in both cases (one less variable is selected when using BIC as the model selection criterion and three less variables are selected when using *Cp* as the model selection criterion). Compared with models using *Cp* as the model selection criterion, models selected by BIC are much more parsimonious.

Table 5.4: Optimal Models without Considering Cost Using BIC as the Model Selection Criterion.

Algorithm	Estimated Coefficients									
based on	Age	Sex	Mstatus	CoMorbid	Duration	ACTmonth	Medtype	Contacts	Jail	MEDCscore
Lasso	0	0	0	0	0	-0.0061	0	0.0028	0	0
Adaptive Lasso	0	0	0	0	0	-0.0088	0	0	0	0
	WAIscore	PSEscore	EMPscore	DAIscore	Lifetime	EmpSC	SubSC	FunSC	DACTS	
Lasso	0	0	0	0	0	0	0.068	0.16	0	
Adaptive Lasso	0	0	0	0	0	0	0.073	0.19	0	

Table 5.5: Optimal Models without Considering Cost Using *Cp* as the Model Selection Criterion.

Algorithm	Estimated Coefficients									
based on	Age	Sex	Mstatus	CoMorbid	Duration	ACTmonth	Medtype	Contacts	Jail	MEDCscore
Lasso	0	0	0	0	0	-0.014	0	0.0070	-0.11	0.086
Adaptive Lasso	0	0	0	0	0	-0.016	0	0.0074	0	0.075
	WAIscore	PSEscore	EMPscore	DAIscore	Lifetime	EmpSC	SubSC	FunSC	DACTS	
Lasso	-0.057	0	-0.32	-0.026	0	0.17	0.10	0.18	0.33	
Adaptive Lasso	0	0	-0.38	0	0	0.22	0.12	0.21	0.49	

5.4.2 Analysis with Cost of Predictors Considered

We now use non-additive costs as shown in Table 5.3. We estimate the regression coefficients to minimize the total loss which is the sum of the residual sum of squares, the lasso type pentalty, and the cost of collecting data for the predictors. The optimization problem has been shown in Equation 3.2.1. Since the adaptive lasso can be solved by LARS using a transformation to the design matrix, we can further adjust the lasso type penalty by using adaptive weights to penalize different coefficients so that the first two parts of the loss function compose the adaptive lasso loss. This optimization problem P can be written as

$$min\ f(\boldsymbol{\beta}, \boldsymbol{\alpha}) = \left\| \mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j \right\|^2 + \lambda \sum_{j=1}^{p} \hat{w}_j |\beta_j| + n\gamma C(\alpha_1 c_1, \ldots, \alpha_p c_p), \qquad (5.4.1)$$

with domain D:

$$\begin{aligned} \alpha_j &\in \{0, 1\},\ for\, j = 1, \ldots, p, \\ \boldsymbol{\beta} &\in \Re^p, \end{aligned}$$

and constraints S:

$$\alpha_j = 0 \ \Rightarrow\ \beta_j = 0,\ for\, j = 1, \ldots, p,$$

The known weights vector $\hat{\mathbf{w}} = (\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_p)^T$ are data-dependent, where $\hat{w}_j, j = 1, \ldots, p$, is the adaptive weight to the regression coefficient β_j. We use the form $\hat{\mathbf{w}} = |\hat{\boldsymbol{\beta}}_{(ols)}|^{-1}$ (Zou, 2006) in this thesis.

First, we use BIC for the lasso (Equation 2.2.2) as the tuning parameter and model selection criterion. We solve the original optimization problem as shown in Equation 3.2.1. When we assign 0.1 to γ, there are 4 predictors selected into the BLARS model the same as lasso model ($\gamma = 0$) did in Section 5.4.1: number of months in ACT, average number of contacts per month, CCAR substance use subscale, and CCAR functioning subscale. When γ was increased to 0.2, 3 predictors remained in the BLARS model, where average number of contacts per month was dropped out. When

γ was increased to 0.5, only number of months in ACT remained in the model. For $\gamma = 1.0$, no variable was selected in the BLARS model due to the cost effect, and the best prediction in this case is the grand mean of the response. The BLARS models for different γ values are shown in Table 5.6. Table 5.8 gives the corresponding Error Sum of Squares and Cost information for these models, where the percentage increase or decrease is compared with the first model ($\gamma = 0.1$).

We then solve the optimization problem as shown in Equation 5.4.1, where BLARS is based on adaptive lasso. When we assign 0.2 to γ, three predictors (number of months in ACT, CCAR substance use subscale, and CCAR functioning subscale) were selected in the BLARS model, the same as adaptive lasso model ($\gamma = 0$) did in Section 5.4.1. For $\gamma = 0.5$, only number of months in ACT remained in the BLARS model, and when γ was increased to 1.0, the best prediction is the grand mean of the response. Table 5.7 displays the BLARS models for different γ values, where BLARS is based on adaptive lasso. Table 5.9 gives the corresponding Error Sum of Squares and Cost information for these models, where the percentage increase or decrease is compared with the first model ($\gamma = 0.2$).

Table 5.6: Optimal Models Using BIC as the Model Selection Criterion. BLARS based on Lasso.

					Estimated Coefficients					
γ	Age	Sex	Mstatus	CoMorbid	Duration	ACTmonth	Medtype	Contacts	Jail	MEDCscore
0.1	0	0	0	0	0	-0.011	0	0.0074	0	0
0.2	0	0	0	0	0	-0.010	0	0	0	0
0.5	0	0	0	0	0	-0.015	0	0	0	0
1.0	0	0	0	0	0	0	0	0	0	0
γ	WAIscore	PSEscore	EMPscore	DAIscore	Lifetime	EmpSC	SubSC	FunSC	DACTS	
0.1	0	0	0	0	0	0	0.11	0.20	0	
0.2	0	0	0	0	0	0	0.10	0.21	0	
0.5	0	0	0	0	0	0	0	0	0	
1.0	0	0	0	0	0	0	0	0	0	

Table 5.7: Optimal Models Using BIC as the Model Selection Criterion. BLARS based on Adaptive Lasso.

	Estimated Coefficients									
γ	Age	Sex	Mstatus	CoMorbid	Duration	ACTmonth	Medtype	Contacts	Jail	MEDCscore
0.2	0	0	0	0	0	-0.013	0	0	0	0
0.5	0	0	0	0	0	-0.019	0	0	0	0
1.0	0	0	0	0	0	0	0	0	0	0
γ	WAIscore	PSEscore	EMPscore	DAIscore	Lifetime	EmpSC	SubSC	FunSC	DACTS	
0.2	0	0	0	0	0	0	0.12	0.23	0	
0.5	0	0	0	0	0	0	0	0	0	
1.0	0	0	0	0	0	0	0	0	0	

Table 5.8: Objective Values Using BIC as the Model Selection Criterion. BLARS based on Lasso.

γ	Total Loss	SSE	SSE Increase	Cost (per patient)	Cost Decrease
0.1	348	317	-	1.15	-
0.2	370	325	2.5%	0.925	19.6%
0.5	384	368	15.8%	0.125	89.1%
1.0	388	388	22.2%	0	-

Table 5.9: Objective Values Using BIC as the Model Selection Criterion. BLARS based on Adaptive Lasso.

γ	Total Loss	SSE	SSE Increase	Cost (per patient)	Cost Decrease
0.2	377	323	-	0.925	-
0.5	382	366	13.2%	0.125	86.5%
1.0	388	388	19.9%	0	-

Second, we use *Cp* (Efron *et al.*, 2004) as the tuning parameter and model selection criterion. We solve the original optimization problem as shown in Equation 3.2.1. When 0.01 was assigned to γ, there were 11 predictors selected in the BLARS model as lasso model ($\gamma = 0$) did in Section 5.4.1: number of months in ACT, average number of contacts per month, ever in jail, adherence to medication scale, working alliance inventory, empowerment scale, drug attitude inventory, CCAR employment subscale, CCAR substance use subscale, CCAR functioning subscale, and Dartmonth ACT scale. When $\gamma = 0.02$, there was one variable (Dartmonth ACT scale) dropped out and the BLARS model contained 10 predictors. When γ was increased to 0.04, two more variables (ever in jail and adherence to medication scale) were dropped out and 8 predictors remained in the BLARS model. When $\gamma = 0.1$, the BLARS model contained 4 variables: number of months in ACT, average number of contacts per month, CCAR substance use subscale and functioning subscale. There were 3 predictors (number of months in ACT, CCAR substance use subscale and functioning subscale) selected for $\gamma = 0.2$. When γ was increased to 0.5, only the number of months in ACT remained in the model. For $\gamma = 1.0$, no variable was selected in the BLARS model due to the cost effect, and the best prediction was the grand mean of the response. The BLARS models for different γ values are displayed in Table 5.10. Table 5.12 gives the corresponding Error Sum of Squares and Cost information for these models, where the percentage increases or decreases are compared with the first model ($\gamma = 0.01$).

We also solve the optimization problem as shown in Equation 5.4.1, where BLARS is based on adaptive lasso. When $\gamma = 0.005$, BLARS selected 8 variables as adaptive lasso model ($\gamma = 0$) did in Section 5.4.1. When we assigned 0.02 to γ, there was one variable (Dartmonth ACT scale) dropped out. Adherence to medication scale was dropped out for $\gamma = 0.03$, and CCAR employment subscale was dropped out for $\gamma = 0.04$. For $\gamma = 0.1$, BLARS selected 4 variables. When γ was increased to 0.2, one less variable was selected, where average number of contacts per month was dropped out. For $\gamma = 0.5$, only number of months in ACT remained in the BLARS model, and when γ was increased to 1.0, the best prediction was the grand mean of the response. Table 5.11 displays the BLARS models for different γ values, where BLARS is based on adaptive lasso. Table 5.9 gives the corresponding Error Sum of Squares and Cost information for these models, where the percentage increases or decreases are compared with the first model ($\gamma = 0.005$). Figure 5.1 shows the search tree of the optimal BlARS model with $\gamma = 0.02$. The red path is the chosen optimal path.

Table 5.10: Optimal Models Using *Cp* as the Model Selection Criterion. BLARS based on Lasso.

	Estimated Coefficients									
γ	Age	Sex	Mstatus	CoMorbid	Duration	ACTmonth	Medtype	Contacts	Jail	MEDCscore
0.01	0	0	0	0	0	-0.014	0	0.0070	-0.11	0.086
0.02	0	0	0	0	0	-0.012	0	0.0069	-0.15	0.091
0.04	0	0	0	0	0	-0.012	0	0.0067	0	0
0.10	0	0	0	0	0	-0.011	0	0.0074	0	0
0.20	0	0	0	0	0	-0.010	0	0	0	0
0.50	0	0	0	0	0	-0.015	0	0	0	0
1.00	0	0	0	0	0	0	0	0	0	0
γ	WAIscore	PSEscore	EMPscore	DAIscore	Lifetime	EmpSC	SubSC	FunSC	DACTS	
0.01	-0.057	0	-0.32	-0.026	0	0.17	0.096	0.18	0.33	
0.02	-0.060	0	-0.35	-0.046	0	0.17	0.090	0.18	0	
0.04	-0.055	0	-0.29	-0.070	0	0.17	0.100	0.19	0	
0.10	0	0	0	0	0	0	0.112	0.20	0	
0.20	0	0	0	0	0	0	0.101	0.21	0	
0.50	0	0	0	0	0	0	0	0	0	
1.00	0	0	0	0	0	0	0	0	0	

Table 5.11: Optimal Models Using *Cp* as the Model Selection Criterion. BLARS based on Adaptive Lasso.

	Estimated Coefficients									
γ	Age	Sex	Mstatus	CoMorbid	Duration	ACTmonth	Medtype	Contacts	Jail	MEDCscore
0.005	0	0	0	0	0	-0.016	0	0.0074	0	0.075
0.02	0	0	0	0	0	-0.014	0	0.0071	0	0.085
0.03	0	0	0	0	0	-0.013	0	0.0074	0	0
0.04	0	0	0	0	0	-0.013	0	0.0074	0	0
0.10	0	0	0	0	0	-0.013	0	0.0080	0	0
0.20	0	0	0	0	0	-0.013	0	0	0	0
0.50	0	0	0	0	0	-0.019	0	0	0	0
1.00	0	0	0	0	0	0	0	0	0	0
γ	WAIscore	PSEscore	EMPscore	DAIscore	Lifetime	EmpSC	SubSC	FunSC	DACTS	
0.005	0	0	-0.38	0	0	0.22	0.12	0.21	0.49	
0.02	0	0	-0.43	0	0	0.21	0.11	0.21	0	
0.03	0	0	-0.41	0	0	0.24	0.12	0.22	0	
0.04	0	0	-0.43	0	0	0	0.12	0.22	0	
0.10	0	0	0	0	0	0	0.12	0.21	0	
0.20	0	0	0	0	0	0	0.12	0.23	0	
0.50	0	0	0	0	0	0	0	0	0	
1.00	0	0	0	0	0	0	0	0	0	

Table 5.12: Objective Values Using *Cp* as the Model Selection Criterion. BLARS based on Lasso.

γ	Total Loss	SSE	SSE Increase	Cost (per patient)	Cost Decrease
0.01	322	301	-	2.95	-
0.02	325	303	0.9%	2.15	27.1%
0.04	335	308	2.5%	1.65	44.1%
0.10	348	317	5.5%	1.15	61.0%
0.20	370	325	8.2%	0.925	68.6%
0.50	384	368	22.2%	0.125	95.8%
1.00	388	388	28.9%	0	-

When γ is small, BLARS models based on adaptive lasso are more parsimonious compared with the BLARS models based on lasso using either BIC or *Cp* as the turning parameter and model selection criterion. Compared with models using *Cp* as the model selection criterion, models selected by BIC are much more parsimonious for small values of γ. However when γ is getting bigger ($\gamma > 0.1$), BLARS results based on lasso are similar as the BLARS results based on adaptive lasso regardless which model selection criterion is used.

The value of γ is user-defined, and the selection criteria of tuning parameter and model selection are also user's choice. The health researchers, or decision makers, should make overall judgements based on the percentage increase of the error sum of squares and the percentage decrease of the cost to choose their preferred cost-efficient model from the BLARS results. Note that in this chapter, the variables are selected and their effects are estimated in the purpose of prediction. They do not necessarily explain the causal effects.

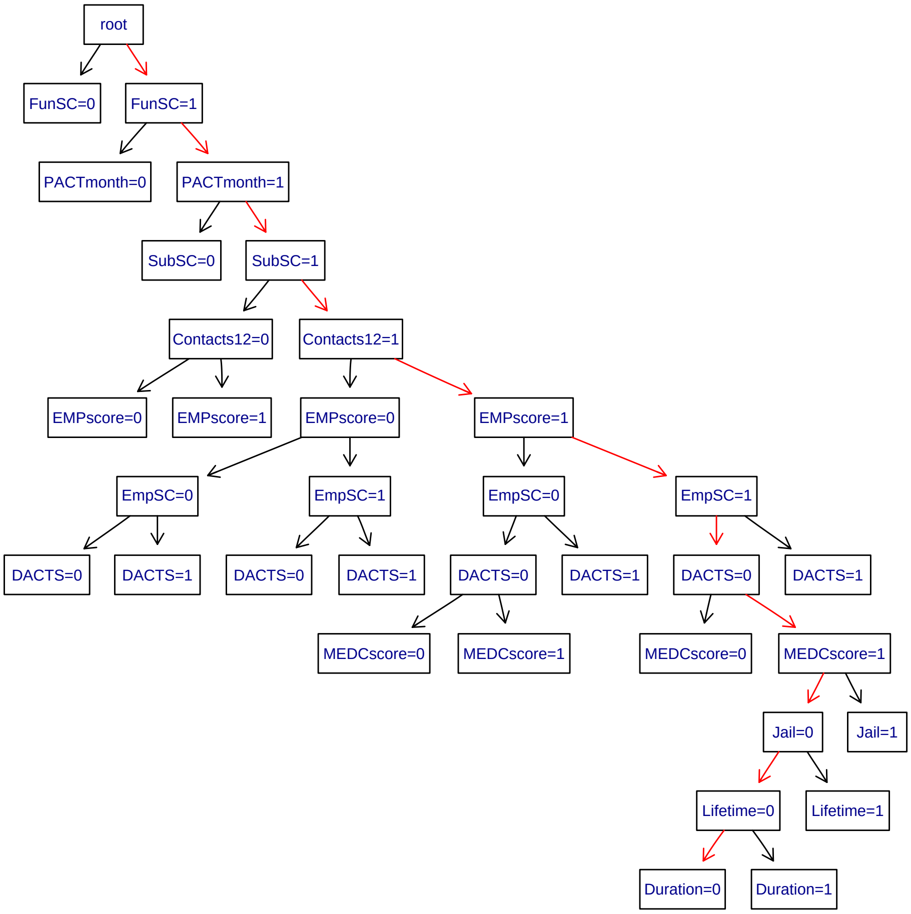

Figure 5.1: Search Tree for Optimal BLARS Model with $\gamma = 0.02$ Using Cp as the Model Selection Criterion. BLARS is Based on Adaptive Lasso.

Table 5.13: Objective Values Using *Cp* as the Model Selection Criterion. BLARS based on Adaptive Lasso.

γ	Total Loss	SSE	SSE Increase	Cost (per patient)	Cost Decrease
0.005	335	300	-	2.7	-
0.02	340	305	1.6%	1.9	29.6%
0.03	342	307	2.3%	1.65	38.9%
0.04	345	311	3.6%	1.45	46.3%
0.10	354	316	5.2%	1.15	57.4%
0.20	377	323	7.7%	0.925	65.7%
0.50	382	366	21.9%	0.125	95.4%
1.00	388	388	29.1%	0	-

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

This thesis was motivated by the Assertive Community Treatment (ACT) project I was involved in, where the cost of collecting data for the potential predictors were different due to different sources of data collection. In this thesis, we have developed a variable selection procedure called Branching LARS (BLARS) that can simultaneously select and estimate the important predictors to build a model that is not only good at prediction but also cost efficient. The BLARS optimization problem is an extension of the lasso to incorporate variable costs penalized in the objective loss function, and a modified branch and bound method is employed to search for a model which minimizes total loss. The total loss includes the residual sum of squares, the lasso type penalty, and the cost of collecting data for the predictors, where the first two parts compose the lasso loss. We can further adjust BLARS by using an adaptive lasso type penalty instead of a lasso type penalty. We focused on lasso type penalty when we were developing the BLARS procedure since the R package *lars* can be directly used to solve lasso type optimization problem.

Additive cost is the simplest cost structure, where the total cost of obtaining data for a selected set of variables is the sum of the cost of getting data for each variable in the set. This cost structure applies to the situation when we collect the data for the variables individually and independently. Although additive cost only exists in some special cases, we developed the BLARS algorithm based on additive cost as a starting point because of its simplicity. More general cost structure is non-additive cost, where a grouping effect of cost is a common example.

We also built an R package *branchLars* which implements the BLARS procedure. This software implementation makes BLARS more practical, and the health researchers can routinely apply the cost-efficient variable selection strategy in statistical model building given they have the cost information at hand. The diabetes data from Efron *et al.*(2004) was used in the examples to illustrate the development of BLARS procedure and the usage of the R package *branchLars*, and BIC for the lasso (Equation 2.2.2) was used in the selection of the turning parameter and as model selection criterion for its simplicity and effectiveness.

BLARS was applied in the data analysis of the ACT project. In the analysis, we further adjusted the lasso type penalty by using adaptive weights to penalize different coefficients so that the first two parts of the loss function compose the adaptive lasso loss. The cost-efficient variable selection results based on both lasso type penalty and adaptive lasso type penalty were shown in details. Both BIC for the lasso (Zou *et al.*, 2007) and Cp (Efron *et al.*, 2004) were used in the selection of the turning parameter and as model selection criteria, and we made comparisons for the results.

6.2 Discussion

Our cost efficient variable selection method is based on the LARS technique, which may not be suitable in some nonlinear situations. We could generalize our BLARS algorithm by changing the Lasso loss (the first two terms in equation (2.2.1)) to other minimization objective functions to incorporate the cost effect whenever we have a method to solve that minimization problem. Recently Friedman *et al.* (2008) proposed new fast algorithms for regression estimation which are based on cyclical coordinate descent methods. Their methods are a remarkably fast approach for solving convex problems with an l_1 (the lasso) penalty or l_2 (the ridge-regression) penalty, or mixtures of the two (the elastic-net penalty). Since these alternatives are well developed, they can be easily adapted to the node-level in our cost efficient variable searching approach, but unfortunately they are not directly applicable to minimizing Equation (3.2.1), which is not convex.

We illustrated the cost-efficient variable selection procedure in this thesis with either BIC for the lasso or Cp as the turning parameter and model selection criteria. There is a lot of controversy on which criterion is the best, and it seems that no one surpasses others in all situations. Researchers may have their preferred selection criteria other than BIC or Cp, and they have to make the judgement based on their own opinion. But the BLARS algorithm is the same regardless which model selection criterion is used.

6.3 Future Work

We considered two cost components, monetary cost and level of difficulty, in the ACT project. Because the two components were estimated in the same scale, we used the combined overall costs in the data analysis. In general cases, the two cost components may not be in the same scale, therefore it may be better to consider them separately, and it gives researchers more freedom to balance between the two kind of costs.

Similar to the Equation 3.2.1, but with two separate cost components, the optimization problem P can be written as

$$\begin{aligned} min\ f(\boldsymbol{\beta}, \boldsymbol{\alpha}) = \left\| \mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j \right\|^2 + \lambda \sum_{j=1}^{p} |\beta_j| \quad &+ \quad n\gamma_1 C_1(\alpha_1 c_1, \ldots, \alpha_p c_p) \\ &+ \quad n\gamma_2 C_2(\alpha_1 c_1, \ldots, \alpha_p c_p), \end{aligned}$$

with domain D:

$$\begin{aligned} \alpha_j &\in \{0, 1\},\ for\ j = 1, \ldots, p, \\ \boldsymbol{\beta} &\in \Re^p, \end{aligned}$$

and constraints S:

$$\alpha_j = 0 \ \Rightarrow\ \beta_j = 0,\ for\ j = 1, \ldots, p,$$

where $\boldsymbol{\beta}$ is the regression coefficient vector that we want to estimate; $\lambda \geq 0$ is the regularization or tuning parameter. The cost function $C_1(\alpha_1 c_1, \ldots, \alpha_p c_p)$ represents the total monetary cost and $C_2(\alpha_1 c_1, \ldots, \alpha_p c_p)$ represents the cost reflecting the level of difficulty to collect the data. The cost functions are non-decreasing when adding more $\alpha_k = 1$ to the existing non-zero set of $\{\alpha_j\}$. The two parameters $\gamma_1 \geq 0$ and $\gamma_2 \geq 0$ are two user-defined weights imposed on costs, reflecting the level of reluctance to use high monetary cost variables and variables with high level of collecting difficulties respectively.

In the data analysis of the ACT project, we further adjusted the optimization problem by using an adaptive lasso penalty instead of a lasso type penalty in the objective function (Equation 5.4.1), where the data-dependent weights vector $\hat{\mathbf{w}} = (\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_p)^T$ has the form $|\hat{\beta}_j|^{-\nu}$ for each $\hat{w}_j, j = 1, \ldots, p$. In this thesis, we fixed $\nu = 1$ and used $\hat{\mathbf{w}} = |\hat{\boldsymbol{\beta}}_{(ols)}|^{-1}$, where $\hat{\boldsymbol{\beta}}_{(ols)}$ is the vector of ordinary least squares estimates. This is suitable when collinearity is not a concern. In the future, we can try $\hat{\boldsymbol{\beta}}_{(ridge)}$ from the ridge regression fit when collinearity is a concern because it is more stable than $\hat{\boldsymbol{\beta}}_{(ols)}$ in this case. Another time-consuming but worth-trying future work is to use two-dimensional cross-validation to turn the two parameters λ and ν when the adaptive lasso penalty is used in the objective function.

BIBLIOGRAPHY

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics* **37** (4), 373–384.

Brown, P. J., Fearn, T. and Vannucci, M. (1999). The choice of variables in multivariate regression: a non-conjugate bayesian decision theory approach. *Biometrika* **86** (3), 635–648.

Efron, B., Hastie, T., Johnston, I. and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics* **32** (2), 407–451.

Ellis, R. H., Wilson, N. Z. and Foster, F. M. (1984). Statewide treatment outcome assessment in Colorado: The Colorado Client Assessment Record (CCAR). *Community Mental Health Journal* **20** (1), 72–89.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* **55** (1), 119–139.

Friedman, J., Hastie, T. and Tibshirani, R. (2008). Regularization paths for generalized linear models via coordinate descent. Technical report Department of Statistics, Stanford University.

Hastie, T. and Efron, B. (2007). *lars: Least Angle Regression, Lasso and Forward Stagewise.* R package version 0.9-7.

Hastie, T., Tibshirani, R. and Friedman, J. (2001). *The Elements of Statistical Learning.* Springer-Verlag.

Hesterberg, T., Choi, N. H., Meier, L. and Fraley, C. (2008). Least angle and l_1 penalized regression: a review. *Statistics Surveys* **2**, 61–93.

Hooker, J. N. (2007). *Integrated methods for optimization.* New York: Springer.

Lehman, A. F. and Steinwachs, D. M. (1998). Translating research into practice: the schizophrenia patient outcomes research team (PORT) treatment recommendations. *Schizophrenia Bulletin* **24** (1), 1–10.

Lindley, D. V. (1968). The choice of variables in multiple regression. *Journal of the Royal Statistical Society. Series B (Methodological)* **30** (1), 31–66.

Meinshausen, N. (2007). Relaxed lasso. *Computational Statistics and Data Analysis* **52** (1), 374–393.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics* **34** (3), 1436–1462.

Ministry of Health (1998). *Ontario Standards for Assertive Community Treatment Teams.* Ontario Ministry of Health and Long-Term Care Toronto, Ontario.

Ministry of Health (1999). *Comprehensive Assessment Project.* Ontario Ministry of Health and Long-Term Care Toronto, Ontario.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing Vienna, Austria. ISBN 3-900051-07-0.

Shao, J. (1997). An asymptotic theory for linear model selection. *Statistica Sinica* **7** (2), 221–264.

Teague, G. B., Bond, G. R. and Drake, R. E. (1998). Program fidelity in Assertive Community Treatment: development and use of a measure. *American Journal of Orthopsychiatry* **68**, 216–232.

Teague, G. B., Drake, R. E. and Ackerson, T. H. (1995). Evaluating use of continuous treatment teams for persons with mental illness and substance abuse. *Psychiatric Services* **46** (7), 689–695.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* **58** (1), 267–288.

Wang, H. and Leng, C. (2007). Unified LASSO estimation via least squares approximation. *Journal of the American Statistical Association* **102** (479), 1039–1048.

Wang, H., Li, R. and Tsai, C. (2007). Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika* **93** (3), 553–568.

Yang, Y. (2005). Can the strengths of AIC and BIC be shared? A conflict between model identification and regression estimation. *Biometrika* **92** (4), 937–950.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association* **101** (476), 1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67** (2), 301–320.

Zou, H., Hastie, T. and Tibshirani, R. (2007). On the “degrees of freedom” of the lasso. *The Annals of Statistics* **35** (5), 2173–2192.

VITA

Name

Li Hua Yue

Education

Doctor of Philosophy in Statistics, December 2010
The University of Western Ontario
London, Ontario, Canada
Supervisors: Dr. Duncan Murdoch & Dr. Wenqing He

Bachelor of Science in Statistical & Actuarial Sciences, April 2006
The University of Western Ontario
London, Ontario, Canada

Bachelor of Engineering in Thermo Engineering, July 1996
Northeastern University
Shenyang, Liaoning, China

Experience

Collaborative Research in ACT Project, 2007 - 2010
Department of Statistical & Actuarial Sciences, Department of Family Medicine
The University of Western Ontario

Teaching Assistant, 2006 - 2010
Department of Statistical & Actuarial Sciences, The University of Western Ontario

Graduate Research Assistant, 2006 - 2010
Department of Statistical & Actuarial Sciences, The University of Western Ontario

Undergraduate Research Assistant, 2005 - 2006
Department of Statistical & Actuarial Sciences, The University of Western Ontario
Research Topic: Forecast Forest Fires Using Statistical Models

Instructor, 1996 - 2001
Department of Mechanical & Electronic Engineering
Liaoyang Vocational-Technical Institute, China

Academic Counselor, 1996 - 2001
Department of Mechanical & Electronic Engineering
Liaoyang Vocational-Technical Institute, China

Scholarships and Awards

1. NSERC PGS D
   The University of Western Ontario, 2007-2010

2. Statistical Society of Canada Student Travel Award
   Annual Meeting of the Statistical Society of Canada, 2008

3. NSERC CGS M
   The University of Western Ontario, 2006-2007

4. Governor General's Medal
   The University of Western Ontario, 2006

5. The Frances Weir Scholarship
   The University of Western Ontario, 2006

6. The London Life Insurance Company Gold Medal
   The University of Western Ontario, 2006

7. John A. Mereu Book Prize
   The University of Western Ontario, 2006

8. NSERC Undergraduate Student Research Award (USRA)
   The University of Western Ontario, summer 2005, summer 2006

9. Dean's Honor List
   The University of Western Ontario, 2003, 2004, 2005, 2006

10. USRA Meteorological Supplements
    The University of Western Ontario, 2005