Variable Selection Methods with Applications

by

Junshan Qiu

(Under the direction of Xiangrong Yin)

Abstract

A data-driven method of generalized adaptive ridge (gar) for an automatic yet adaptive regression shrinkage and selection was proposed as the first part of this dissertation. In theory, gar was proved to be equivalent to adaptive- lasso, ridge regression and elastic net under appropriate conditions. Simulation results indicated gar performs either better or equivalent to these methods, in terms of both prediction accuracy and computational speed due to its flexibility of our newly developed algorithm.

The second part of this dissertation is on a general adaptive L^2-regularized optimization problem for a general loss function, ℓ. This adaptive L^2 penalty term was proved to be equivalent to adaptive L^1 penalty, adaptive L^2 penalty, and combined adaptive L^1 and L^2 penalty with appropriate choice of parameters and assuming ℓ is differentiable. Two algorithms using Newton-Raphson method for the case of the number of predictors (p) less than the number of sample size (n), and sequential minimal optimization (SMO) method for the case $p > n$ and correlated data were developed. The efficacy of this approach was illustrated by simulations, comparisons with other methods and real data analysis.

The last part of this dissertation is about adaptive three-way decomposition (ATWD) which combines the adaptive approach in the first part of this dissertation and popular three-way decomposition (TWD) in chemical sciences to analyze nuclear magnetic resonance

(NMR) data. This method can be used to reduce the effects of signal noises and dimensionality of the spectral data, provide efficient estimates of spectral components, interpret all the signals retrieved from NMR data, and translate structural information efficiently. Its effective usefulness were illustrated in both simulation studies and real data analysis.

INDEX WORDS: ADAPTIVE LASSO, ELASTIC NET, GENERALIZED ADAPTIVE RIDGE, GENERALIZED LINEAR MODEL, LASSO, NMR, PENALIZATION, PENALIZED LIKELIHOOD, RIDGE REGRESSION, VARIABLE SELECTION

Variable Selection Methods with Applications

by

Junshan Qiu

B.A., Shenyang Pharmaceutical University, China, 1999

M.S., Institute of Applied Ecology, China, 2002

M.S., University of Georgia, 2004

Ph.D., University of Georgia, 2008

A Dissertation Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2008

Variable Selection Methods with Applications

by

Junshan Qiu

Approved:

Major Professor: Xiangrong Yin

Committee: Yehua Li
William P. McCormick
Jaxk Reeves
Lynne Seymour

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
December 2008

Dedication

To my family

Acknowledgments

First, I would like to thank my family, without their support and love I could not finish the second Ph.D. degree in my life.

Secondly, I would like to thank my major advisor, Dr. Xiangrong Yin, for guiding me through darkness and hardness before I reach the final goal.

I would also like to acknowledge Drs. Li, McCormick, Reeves and Seymour for serving on my committee. Dr. Reeves also deserves a special thank for helping in writing.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction and literature review

1.1 Introduction

Since early 1970s, there has been considerable literature on variable selection. Variable selection has become an essential part of statistical analysis and has received more and more attention in recent years. For instance, often in bioinformatics or longitudinal studies, there are many variables measured. In order to enhance model predictability and model parsimony, it is common in practice to include only a subset of important variables in the model. We will give a short overview on some representative methods.

In section 1.2.1, we focus on variable selection methods for the linear model and mainly on methods based on penalized least squares approach. In section 1.2.2, we review variable selection methods for generalized linear models. In section 1.2.3 we give some detail on typical backward and forward algorithms. Microarray, as a new technology to investigate expression levels of thousands of genes simultaneously, has brought many statistical problems. One of them is to reduce the dimension of microarray data and select important variables such as types of genes. Many statistical methods have been developed to solve this problem. In section 1.2.4, we give a short overview of variable selection methods related to microarray data analysis. Nuclear Magnetic Resonance (NMR), as a powerful technique to provide information on structural and chemical properties of molecules, has become a new area which draws more attention of statisticians. NMR data also has high dimensions. Signal noises sometimes can affect analysis results significantly. To find the true structures of chemicals, dimension reduction and variable selection methods will be useful tools to achieve that goal. In section 1.2.5, we present a review on typical methods related to NMR data analysis

proposed. Finally, in section 1.2.6 we outline our proposed dissertation of developing new variable selection methods and applications on microarray data and NMR data.

1.2 Literature Review

Throughout this presentation, we denote Y_i as the scalar response collected from the ith $(1 \leq i \leq n)$ subject and let $X_i = (X_{i1}, \cdots, X_{ip})^\top \in \mathbb{R}^p$ be the associated p-dimensional predictor and pre-normalized with mean zero and unit variance.

1.2.1 Variable selection for linear model

In this section, we summarize some penalized least square methods for variable selection. To model the regression relationship between Y_i and X_i, the following linear regression model is typically considered

$$Y_i = X_i^\top \beta + \epsilon_i, \tag{1.1}$$

where $\beta = (\beta_1, \cdots, \beta_p)^\top \in \mathbb{R}^p$ is the regression coefficient vector and ϵ_i is the random noise. To obtain a sensible estimator for β, the following ordinary least squares (OLS) estimator has been extensively used

$$\hat{\beta}^{\text{OLS}} = \text{argmin}_\beta \|Y - X\beta\|^2,$$

where $Y = (Y_1, \cdots, Y_n)^\top \in \mathbb{R}^n$ is the response vector, $X = (X_1, \cdots, X_n)^\top \in \mathbb{R}^{n \times p}$ is the design matrix, and $\|\cdot\|$ stands for the typical L_2 norm.

However, as noted by many researchers, the OLS estimator $\hat{\beta}^{\text{OLS}}$ suffers quite a number of important limitations. For example, its finite sample variance can be very large if the design matrix X suffers from the problem of multi-collinearity. Furthermore, it is not directly applicable if the predictor dimension p is larger than the sample size n. As a simple yet neat solution, Hoerl and Kennard (1970a,b) proposed the following ridge estimator

$$\hat{\beta}_\lambda^{\text{RIDGE}} = \text{argmin}_\beta \Big\{ \|Y - X\beta\|^2 + \lambda \|\beta\|^2 \Big\},$$

where $\lambda > 0$ is a tuning parameter, which controls the amount of the penalty applied to the regression coefficient. As one can see, the ridge estimator $\hat{\beta}_{\lambda}^{\text{RIDGE}}$ is nothing but a L_2-penalized least squares estimator. Furthermore, it is also a Bayesian estimator with a multivariate Gaussian prior placed on the regression coefficient β. Hoerl and Kennard (1970a, b) showed that there always exist a λ value so that the ridge estimator $\hat{\beta}_{\lambda}^{\text{RIDGE}}$ is more accurate than the OLS estimator $\hat{\beta}^{\text{OLS}}$ in terms of the mean squared error (MSE). Consequently, $\hat{\beta}_{\lambda}^{\text{RIDGE}}$ has been well recognized for its high prediction accuracy. In addition to that, one can find that $\hat{\beta}_{\lambda}^{\text{RIDGE}}$ is computable even if the design matrix X is singular (e.g., severe multi-collinearity or too large predictor dimension $p \gg n$). All those merits together make $\hat{\beta}_{\lambda}^{\text{RIDGE}}$ a practically very useful regression estimator.

Nevertheless, the ridge estimator $\hat{\beta}_{\lambda}^{\text{RIDGE}}$ also suffers some limitations. In particular, it does not have the capability to permit variable selection. The reason is that $\hat{\beta}_{\lambda}^{\text{RIDGE}}$, similar to the OLS estimator $\hat{\beta}^{\text{OLS}}$, cannot produce parsimonious solution for the estimated regression coefficient. As a result, many irrelevant predictors are included in the model, which naturally deteriorates the prediction accuracy and hurts the practical interpretability. As a nice solution, Tibshirani (1996) proposed the LASSO (least absolute shrinkage and selection operator) estimator, which is given by

$$\hat{\beta}_{\gamma}^{\text{LASSO}} = \text{argmin}_{\beta}\Big\{ \|Y - X\beta\|^2 + \gamma|\beta| \Big\},$$

where $\gamma > 0$ is once again the tuning parameter and $|\beta|$ is defined to be $|\beta| = \sum |\beta_j|$. As one can see, the lasso estimator $\hat{\beta}_{\gamma}^{\text{LASSO}}$ is nothing but a L_1-penalized estimator. The major strength of the lasso estimator is that $\hat{\beta}_{\gamma}^{\text{LASSO}}$ has the capability to produce parsimonious solutions in its estimated regression coefficient. Consequently, those variables associated with the parsimonious solutions are automatically identified as relevant variables. By doing so, $\hat{\beta}_{\gamma}^{\text{LASSO}}$ elegantly combines parameter estimation and variable selection as one unified task.

Unfortunately, LASSO also suffers its own limitations. In particular, LASSO cannot handle situations with serious multi-collinearity or $p \gg n$ satisfactorily (Zou and Hastie, 2005). Although it has been repeatedly observed that LASSO can indeed improve the estimation

efficiency (hence, prediction accuracy), such an improvement is mainly due to the fact that irrelevant variables are detected and removed. In an extreme situation (for example), where $\beta_j > 0$ for every $1 \leq j \leq p$, it can be shown easily that, with probability tending to one, the lasso estimator $\hat{\beta}_\gamma^{\text{LASSO}}$ can be expressed as $\hat{\beta}_\gamma^{\text{LASSO}} = \hat{\beta}^{\text{OLS}} + \gamma \times \mathbf{1}$ (Efron, Hastie, Johnstone and Tibshirani 2004; Zou, Hastie and Tibshirani 2007), where $\mathbf{1} = (1, \cdots, 1) \in \mathbb{R}^p$ is a constant vector. Such a simple formula reveals immediately that $\hat{\beta}_\gamma^{\text{LASSO}}$ produces even worse estimation efficiency than $\hat{\beta}^{\text{OLS}}$. This is because $\hat{\beta}_\gamma^{\text{LASSO}}$ shares the same variance as $\hat{\beta}^{\text{OLS}}$ but suffers an extra bias due to the $\gamma \times \mathbf{1}$ term. However, it is noteworthy to remark that this is a situation for which the ridge estimator $\hat{\beta}_\lambda^{\text{RIDGE}}$ can help, particularly if X suffers the problem of multi-collinearity (Zou and Hastie, 2005). Thus, a natural yet interesting question arises: can we combine the strengths of the ridge regression and LASSO methods together?

As an elegant solution, Zou and Hastie (2005) proposed the method of ELASTIC NET, which estimate the regression coefficient by

$$\hat{\beta}_\theta^{\text{ENET}} = \text{argmin}_\beta \Big\{ \|Y - X\beta\|^2 + \lambda \|\beta\|^2 + \gamma |\beta| \Big\},$$

where $\theta = (\lambda, \gamma)$ is the tuning parameter vector. Compared with the ridge estimator $\hat{\beta}_\lambda^{\text{RIDGE}}$, the ELASTIC NET estimator $\hat{\beta}_\theta^{\text{ENET}}$ is able to do variable selection. Compared with the LASSO estimator $\hat{\beta}_\gamma^{\text{LASSO}}$, the elastic net estimator $\hat{\beta}_\theta^{\text{ENET}}$ can deal with multi-collinearity and $p \gg n$ problems. For such a reason, the $\hat{\beta}_\theta^{\text{ENET}}$ has achieved quite a success in both theory and application.

1.2.2 Variable selection for generalized linear model

Usually, parameter estimation for generalized linear models is based on a likelihood function. The idea of penalized least square method can be extended to penalized likelihood and can be extended to any loss function naturally. A general form of penalized likelihood proposed

by Fan and Li (2001) can be formulated as follows:

$$\ell(\beta) + n \sum_{j=1}^{p} p_\lambda(|\beta|), \tag{1.2}$$

where ℓ is a loss function and $p_\lambda(|\beta|)$ is a penalty function. The loss function could be in least-squares form or in likelihood form. The forms of penalty functions are also diverse. They could be L_1 penalty, L_2 penalty, L_1 and L_2 combinations, functions of L_1 or L_2 and higher order penalty. Selecting a suitable penalty function can help statisticians to improve the predictive ability of the model and engineers to distinguish signals from noises.

1.2.3 Algorithms for variable selection

There are two types of algorithms for variable selection: backward and forward. Among them we discuss three representative backward and three representative forward algorithms. For backward algorithms,the first one is $\hat{\beta}_\gamma^{\text{LASSO}}$ (Tibshirani, 1996), the second one is shooting algorithm (Fu, 1998) and the third one is unified algorithm (Fan and Li, 2001). For forward algorithms, three representatives are LARS (Efron et al., 2004), elastic net (Zou and Hastie, 2005), and GLM paths (Park and Hastie, 2007).

Algorithm for lasso

Let $SR(\beta) = \| \mathbf{Y} - X\beta \|^2$, let γ_i, $i = 1, 2, \cdots, 2^p$ be the p-tuples of the form $(\pm 1, \pm 1, \cdots, \pm 1)$, set $R = \{i, \gamma_i \beta = t\}$ and $T = \{i, \gamma_i \beta < t\}$. Here $t \geq 0$ is a tuning parameter. The algorithm proposed by Tibshirani (1996) can be summarized in the following steps.

1. Get OLS estimates $\hat{\beta}_0$ with full model, let $i_0 = sign(\beta_0)$ and $R = \{i_0\}$.

2. Obtain $\hat{\beta}$ by minimizing $SR(\beta)$ with the constraint $R\beta \leq t\mathbf{1}$. Here $\mathbf{1}$ is the one vector.

3. If $\sum_{j=1}^{p} |\beta_j| \leq t$, then Stop. Otherwise go to next step.

4. Put $i = sign(\hat{\beta})$ into R and repeat the process from step 2.

This algorithm performs as a variable selection operator and also keeps the stability of ridge regression. Due to the diversity of penalty functions, it is necessary to develop an algorithm which can be applied to all types of penalty functions.

Shooting algorithm

Another algorithm proposed by Fu (1998) is a general approach which is applicable to any conditions with $\gamma \geq 1$. For a linear model (2.1) with penalty function as $\lambda \sum_{j=1}^{p} |\beta_j|^{\gamma}$, let RSS be the residual sum of squared errors and $D_j(\beta, X, y) = \partial RSS / \partial \beta_j$. Then the shooting algorithm can be summarized as follows.

1. Use $\hat{\beta_{ols}}$ as $\hat{\beta}_0$.

2. At step n, let $D_0 = D_j(0, \hat{\beta^{-j}}, X, y)$, for each $j = 1, \cdots, p$. If $D_0 = 0$, then $\hat{\beta^j} = 0$. For lasso type penalty ($\gamma = 1$), each $\hat{\beta^j}$ can be calculated as follows.

$$\hat{\beta}_j = \frac{\lambda - D_0}{2x_j^T x_j}, \text{ if } D_0 > \lambda; \ \hat{\beta}_j = \frac{-\lambda - D_0}{2x_j^T x_j}, \text{ if } D_0 < -\lambda; \ \hat{\beta}_j = 0, \text{ if } |D_0| \leq \lambda,$$

   where x_j is the jth column of design matrix of X. Then obtain new estimates $\hat{\beta}_n$. For lasso type penalty ($\gamma > 1$), each $\hat{\beta}_j$ is calculated via Newton-Raphson method.

3. Repeat step 2 until convergence is reached.

Although the two algorithm above are illustrated with residual sum of squares as the index, they can be extended naturally to deal with likelihood index. Therefore, these two algorithms can be applied to generalized linear model via the iterative reweighted least square (IRLS) procedure. The common points which these algorithms share are that the penalty terms are transformed into two possible conditions for each β_j (negative or positive). Parameter estimations are based on transformed penalty term.

Unified algorithm

Instead of considering the sign of each β_j, the penalty term can be locally approximated by a quadratic function as in the algorithm by Fan and Li (2001). That is,

$$p_\lambda(|\beta_j|) \approx p_\lambda(|\beta_{j0}|) + \frac{1}{2}p'_\lambda(|\beta_{j0}|)(\beta_j^2 - \beta_{j0}^2), \tag{1.3}$$

where β_j is close to β_{j0}. This approximation is true under the assumption that $p_\lambda(|\beta_j|)$ is differentiable except at the point zero. Further, the loss function is assumed to be smooth with respect to β so that it can be locally approximated by a quadratic function. Once loss function and penalty function are approximated by quadratic functions, the optimization problem can be easily solved by Newton-Raphson method.

It should be noted that λ is a tuning parameter in each algorithm and usually estimated by cross validation methods. All three of the algorithms above are backward process and start with full-model. Since the number of variables being deleted in each step may be greater than one, the running speed of these algorithms may be faster. However, if data is high-dimensional, it is hard to start with full model especially when the number of predictors are much larger than the number of observations. Next, we discuss three forward algorithms for variable selections.

LARS

Given a huge data set with a large collection of covariates, LARS is an efficient algorithm to find coefficients for the parsimonious model selected and improve the predictability of the model. In addition, LARS can also provide exact piecewise linear coefficients paths. The detailed algorithm for LARS can be addressed as follows. Here model 2.1 is used as an example for explanation of this algorithm.

1. Start with $\beta_j = 0$ for $j = 1, \cdots, p$.

2. Find x_{j1} which is most correlated with response variable y.

3. In the direction of x_{j1}, take largest step possible to find another predictor, say x_{j2}, which is most correlated with current residual.

4. Proceed at the direction equiangular between these two predictors to find another predictor, say x_{j3}, which is most correlated with current residual.

5. Continue in the "Least Angle Direction", until all variables selected.

At each step, the parameter estimates are obtained by minimizing the least square term with a L_1 penalty $|\beta| = \sum_{j=1}^{p} |\beta_j|$.

ELASTIC NET

ELASTIC NET was developed directly based on LARS. Instead of using only an L_1 penalty as in LARS, an L_2 penalty is combined with the L_1 penalty in ELASTIC NET. Although some modifications were made to improve the speed for high-dimensional data ($p \gg n$) analysis and to handle group effects efficiently, the basic procedure is the same as with LARS.

GLM Path

The predictor-corrector algorithm is implemented in GLM path. This algorithm has been used in mathematics to implement numerical continuation. In order to elaborate this algorithm, the objective function is defined as

$$F(\beta, \lambda) = -L(y; \beta) + \lambda|\beta|, \tag{1.4}$$

where $L(y; \beta)$ is the likelihood function. Further, let $H(\beta, \lambda) = \frac{\partial \ell}{\partial \beta}$, and assume that any component of β is nonzero and the objective function is differentiable with respect to β. The algorithm can be elaborated as follows.

1. Start with all $\beta_j = 0$ for $j = 1, \cdots, p$ and $\lambda_0 = \lambda_{max}$.

2. At step k, select new predictors by decreasing λ_k to λ_{k+1} and calculate $\hat{\beta}^{k+} = \hat{\beta}^k + (\lambda_{k+1} - \lambda_k)\frac{\partial \beta}{\partial \lambda}$.

3. Obtain $\hat{\beta}^{k+1}$ by optimizing the objective function $\ell(\beta, \lambda_{k+1})$ and with $\hat{\beta}^{k+}$ as starting values.

4. Check if nonzero elements of $\hat{\beta}^{k+1}$ (active set) increases. If active set is augmented, then repeat the process. Otherwise, all process stops.

All of the three algorithms elaborated above start with all parameter estimates set at zero. Significant predictors are selected into active sets by different criteria. In this way, handling huge matrixes for high-dimensional data is avoided. Therefore, these algorithms have shown high efficiency in microarray data analysis.

1.2.4 Microarray expression data analysis

Microarray expression data analysis is a broad topic. Here we focus on dimension reduction of microarray expression data. Some statistical multivariate analysis methods have been applied to dimension reduction of microarray expression data. The classical methods include principal component analysis and singular value decomposition (Alter et al. 2000; West et al. 2000). Recently, penalized logistic regression methods (Zhu and Hastie, 2004; Park and Hastie, 2007) have been developed. In addition, penalized logistic regression methods have powerful predicability and can be easily extended to multiple classification cases.

Optimization algorithms for microarray data analysis

Traditionally, maximum likelihood and Newton-Raphson algorithm are often used to solve penalized likelihood problems. One of the limitations of the Newton-Raphson method is the difficulty of inverting a p by p matrix at each iteration step. The cost of computation is huge where $n \ll p$. In order to make the computation feasible, an algorithm called sequential minimal optimization (SMO) was provided by Platt (1998). One advantage of SMO is that the inversion of the huge matrix can be avoided. This algorithm has been extensively used in microarray data analysis.

Variable selection methods used for microarray data analysis

Predictor variables in microarray expression data are usually expression levels for each gene. Therefore, variable selection is equivalent to gene selection. Ideal variable selection methods for microarray data analysis should satisfy following criteria.

1. Make gene selections or group genes selections efficiently.

2. Handle high dimensional matrix and make computation feasible.

The most well-known miroarray expression data were provided by Golub (1999). This is a leukaemia cancer data set. There are 7129 genes and 72 samples observed. Zhu and Hastie (2004) used a penalized logistic regression method to analyze this data for gene classification and selection; Zou and Hastie (2005) used ELASTIC NET to analyze the same data for important genes related to leukaemia cancer considering genes working the same pathway; Park and Hastie (2007) used GLM path to analyze the same data again for the similar purpose. Comparisons between these methods have been enumerated in the literature (Zou and Hastie, 2005; Park and Hastie, 2007). In terms of group gene selections, the ELASTIC NET method is the best for this data set. However, considering the cost of computation, GLM path has an absolute advantage due to the implementation of the SMO algorithm.

1.2.5 NMR data analysis

Usually, NMR data analysis starts with raw time-domain NMR data which is also called free induction decay (FID). FID represents the primary recording of the NMR data in the time domain and consists of a list of positive or negative numbers as a function of time. Time is often regularly spaced. Data values are presented as two types: real and imaginary. First, FID will be processed with multiplication by a multiplier or window function. To increase digital resolution, zeros will be added to the end of the FID. The goal of NMR data analysis is to obtain a spectrum which provides information on molecular structures, dynamic interactions

between atoms, etc.. Typically, time domain data will be transformed into frequency domain data via Fourier Transforms.

There are several typical methods used in NMR data processing. Among them, we discuss linear prediction, maximum entropy reconstruction, and three-way decomposition (TWD) in detail.

Linear prediction (LP)

The principle which the LP extrapolation method relies on is that future signal values can be derived as linear combination of observed data values (Stephenson, D.S., 1988). The coefficients used in LP extrapolation are estimated from measured FID. The direction of extrapolation can be either forward or backward. Backward extrapolation is used while the beginning of FID is damaged; forward extrapolation is used while data is truncated at the end of FID. Followed by Fourier Transform, the NMR data in the time domain can be transformed into data in the frequency domain. Frequency peaks derived after Fourier Transform can be used to characterize systems in a molecular scale.

Suppose a FID signal sampled at time intervals regularly spaced, F_k, has property as follows. $F_k = \sum_{i=1}^{M} \alpha_i F_{k-i}$, where M is the number of components and α_i is the ith forward prediction coefficients. Similarly, we have $F_k = \sum_{i=1}^{M} \beta_i F_{k+i}$, where β_i is the ith backward prediction coefficients. LP method has been widely used to make spectral improvements in case of defective or incomplete time domain data.

Maximum entropy reconstruction

Maximum entropy reconstruction method is another popular method in NMR data analysis. Consider a simple model for one dimensional NMR data.

$$y(t) \quad = \quad \sum_{i=j}^{M} A_j \exp(\frac{-t}{T_j})(\cos(\omega_j t) + i \sin(\omega_j t)) + \epsilon_t, \tag{1.5}$$

where $y(t)$ is the signal observed at time t, M is the number of components, A is amplitude of the signal, T is the parameter related to decay rate, ω is frequency and ϵ is the noise.

Via the conventional forward Fourier Transform, equation (1.5) has an exact formal inverse assuming T and ω are same for each signal, and it is given by:

$$A_j = (1/M)\sum_{t=1}^{M} y(t)\exp(\frac{t}{T})(\cos(\omega t) + i\sin(\omega t))^{-1} + \epsilon_t^{'}, \tag{1.6}$$

where $\epsilon_t^{'}$ is transformed error at time t. Further, the entropy is defined as

$$S = -\sum_{j=1}^{M} p_j \log p_j, \tag{1.7}$$

where $p_j = A_j / \sum_{i=1}^{M} Aj$.

The procedure of this method can be summarized as follows.

1. Construct trial spectra and compute the hypothetical time domain data which can produce that spectra. Here, trial spectra refers to all possible spectra with a degree freedom for each frequency. Corresponding to each trial spectra, there is a hypothetical time domain data which is also called trial data, $\hat{y}(t)$.

2. Test if hypothetical data is consistent with experimental data by computing a χ^2 statistic. That is,

$$\chi^2 = \sum_{i=1}^{M} \mid y(t) - \hat{y}(t) \mid^2 / \sigma^2, \tag{1.8}$$

   where σ is the standard deviation of expected noise.

3. Collect consistent trial spectra which has level of disagreement less than noise level.

4. Select the trial spectra with maximum entropy.

This method is robust and can deal with data without high quality (Schmieder, et al., 1993). There is no assumption related to the shape of signal and no restriction on sampling time. Since a set of candidate spectra data needs to be constructed first, this method requires a huge amount of calculation. Here the entropy function need to be optimized to find best fit spectra. If L_1 or L_2 penalty terms can be induced into entropy functions, then the effects of noise will be significantly reduced.

Three-way decomposition (TWD)

TWD method (Orekhov, et al., 2003) is much more effective in interpretation of three-dimensional or higher dimension NMR data. There are two principles on which TWD relies. The first is that a signal in multi-dimensional FID can be presented as products of one-dimensional vectors. It can be expressed as shown in the following equation:

$$Y(i,j,k) = \sum_{l=1}^{R} A_l^l F1_i^l F2_j^l F3_k^l + \epsilon_{i,j,k}, \tag{1.9}$$

where $Y(i,j,k)$ is one element of three-dimensional experimental data matrix Y with size (I,J,K) and $i = 1,2...I$; $j = 1,2....J$; $k = 1,2...K$. R is the number of components. $F1^l$, $F2^l$, and $F3^l$ are one-dimensional functions and called shapes. Another principle is that the method of decomposition is unique at least within the three dimensions. Least-square minimization has been used to estimate individual shapes and amplitudes of the R components. That procedure can be written as follows:

$$MIN_{F1,F2,F3,A,R} \sum_{i,j,k} (Y(i,j,k) - \sum_{l=1}^{R} A_l^l F1_i^l F2_j^l F3_k^l)^2. \tag{1.10}$$

A penalized least square method has also been proposed by Orekhov (2003):

$$MIN_{F1,F2,F3,A,R} \sum_{i,j,k} (Y(i,j,k) - \sum_{l=1}^{R} A_l^l F1_i^l F2_j^l F3_k^l)^2 + \lambda \sum_{l=1}^{R} (A_l^l)^2. \tag{1.11}$$

Here, λ is regularization factor. The algorithm used to solve this high-dimensional minimization problem is PARAFAC provided by Harshman and Lundy (1984). The TWD method can be applied to non-uniformly sampled data. Therefore, this method can improve the sensitivity of analysis by sampling on data points with higher signal-to-noise ratio.

1.2.6 Summary

Variable selection methods distinguish from each other by different penalty functions. The loss functions used in variable selection are most often least square errors and likelihood

functions. The algorithms used to optimize objective functions are basically divided into two types: forward and backward type as described previously. My dissertation will focus on variable selection for linear model and generalized linear model with a general loss function including likelihood. The first part, detailed in chapter 2, is to develop a new dynamic approach using only L_2 penalty term for linear model. This method is a data driven approach which is equivalent to LASSO, RIDGE REGRESSION, and ELASTIC NET. The second part, detailed in chapter 3, is to extend the approach derived in the first part to general loss functions. Therefore, this approach can be applied to generalized linear models and also to microarray data analysis. The third part, detailed in chapter 4, is to apply the newly developed statistical variable selection methods to NMR data as well as to investigate some new methods that are particularly useful for NMR data analysis.

Chapter 2

Generalized Adaptive Ridge: a Data Driven Approach[1]

[1]Qiu, J., Yin, X., Wang, H. Submitted to *Technometrics,* 4/2008.

Abstract

In this article we proposed a data-driven method of generalized adaptive ridge (GAR) for an automatic yet adaptive regression shrinkage and selection. We show that, in theory, GAR can be equivalent to adaptive LASSO, adaptive RIDGE REGRESSION and adaptive ELASTIC NET under appropriate conditions. Specifically, if the regression parameters truly enjoy a sparse representation, GAR performs like the most recently proposed ADAPTIVE LASSO (Zou, 2006), hence, is able to identify relevant predictors consistently. If the regression parameters are not that sparse, GAR performs like the adaptive RIDGE REGRESSION, which is well-known for its high prediction accuracy and reliability against the multi-collinearity problem. If the predictor dimension is much larger than the sample size, and the parameters are sparse, GAR performs like adaptive ELASTIC NET which is newly suggested in this paper and an extension from ELASTIC NET. Due to its flexibility, GAR performs either better or equivalent to these methods, in terms of prediction accuracy. Simulation results confirm its competitive performance.

Key Words: ADAPTIVE LASSO; ELASTIC NET; GENERALIZED ADAPTIVE RIDGE; LASSO; RIDGE REGRESSION

2.1 Introduction

Let Y_i be the scalar response collected from the ith ($1 \leq i \leq n$) subject and $X_i = (X_{i1}, \cdots, X_{ip})^\top \in \mathbb{R}^p$ be the associated p-dimensional predictor. The following linear regression of Y_i on X_i is typically considered:

$$Y_i = X_i^\top \beta + \epsilon_i, \tag{2.1}$$

where $\beta = (\beta_1, \cdots, \beta_p)^\top \in \mathbb{R}^p$ is the regression coefficient vector and ϵ_i is the random noise with zero mean and unit variance. The ordinary least squares (OLS) estimator has been extensively used to estimate β.

$$\hat{\beta}^{\text{OLS}} = \text{argmin}_\beta \|Y - X\beta\|^2,$$

where $Y = (Y_1, \cdots, Y_n)^\top \in \mathbb{R}^n$ is the response vector, $X = (X_1, \cdots, X_n)^\top \in \mathbb{R}^{n \times p}$ is the design matrix, and $\|\cdot\|$ stands for the typical L_2 norm.

However, the OLS estimator $\hat{\beta}^{\text{OLS}}$ suffers a number of limitations. For example, its finite sample variance can be very large if the design matrix X suffers from the problem of multi-collinearity. And it is not directly applicable if $p > n$. Hoerl and Kennard (1970a,b) then proposed the following ridge estimator

$$\hat{\beta}_\lambda^{\text{RIDGE}} = \text{argmin}_\beta \Big\{ \|Y - X\beta\|^2 + \tau_1 \|\beta\|^2 \Big\},$$

where $\tau_1 > 0$ is a tuning parameter, which controls the amount of the penalty applied to the regression coefficient. They showed that there always exist a τ_1 value so that the ridge estimator $\hat{\beta}_\lambda^{\text{RIDGE}}$ is more accurate than the OLS estimator $\hat{\beta}^{\text{OLS}}$ in terms of the mean squared error (MSE). Consequently, $\hat{\beta}_\lambda^{\text{RIDGE}}$ has been well recognized for its high prediction accuracy. In addition, $\hat{\beta}_\lambda^{\text{RIDGE}}$ is computable even if the design matrix X is singular (e.g., severe multi-collinearity or too large predictor dimension $p \gg n$). All those merits together make $\hat{\beta}_\lambda^{\text{RIDGE}}$ a practically very useful regression estimator.

However, the ridge estimator $\hat{\beta}_\lambda^{\text{RIDGE}}$ also suffers its own limitation. In particular, it does not have the capability to do variable selection. The reason is that $\hat{\beta}_\lambda^{\text{RIDGE}}$, similar to the OLS estimator $\hat{\beta}^{\text{OLS}}$, cannot produce sparse solution for the estimated regression coefficient. Thus many irrelevant predictors are included in the model, which naturally deteriorates the prediction accuracy and hurts the practical interpretability. To overcome this, Tibshirani (1996) proposed the least absolute shrinkage and selection operator (LASSO) estimator

$$\hat{\beta}_\gamma^{\text{LASSO}} = \text{argmin}_\beta \Big\{ \|Y - X\beta\|^2 + \tau_2 |\beta| \Big\},$$

where $\tau_2 > 0$ is the tuning parameter and $|\beta|$ is defined to be $|\beta| = \sum |\beta_j|$. The main strength of the LASSO estimator is that $\hat{\beta}_\gamma^{\text{LASSO}}$ has the capability to produce sparse solutions in its estimated regression coefficient. Consequently, only those variables associated with the response are automatically identified as important variables. On the other hand, Fan and Li (2001) first noted that the traditional LASSO estimator also suffers a number of limitations.

Specifically, it cannot guarantee a consistent identification of the finite dimensional true model. Furthermore, its estimator cannot be as efficient as the oracle (i.e., the unpenalized estimator obtained under the true model). Such a conjecture are then formally confirmed by Leng, Lin and Wahba (2006), Zou (2006), Yuan and Lin (2007), and Zhao and Yu (2007). As a solution, Zou (2006) then proposed the following adaptive LASSO estimator

$$\hat{\beta}_{\gamma}^{\text{ALASSO}} = \text{argmin}_{\beta}\Big\{\|Y - X\beta\|^2 + \tau_3 \sum_{j=1}^{p} w_j|\beta_j|\Big\},$$

where $\tau_3 > 0$ and w_j's are nonnegative weights for $j = 1, ..., p$. Similar idea was also independently developed for the least absolute deviation regression (Wang, Li and Jiang, 2007), regression with autoregressive errors (Wang, Li and Tsai, 2007), and Cox's proportional hazard model (Zhang and Lu, 2007). The key idea of the adaptive LASSO is to allow for different amount of shrinkage for different coefficients. Intuitively, if larger amounts of shrinkage is used for irrelevant predictors and smaller amount is used for relevant ones, an estimator with better efficiency can be obtained.

Nevertheless, ADAPTIVE LASSO and LASSO cannot handle the situation with serious multi-collinearity and $p \gg n$. As an elegant solution, Zou and Hastie (2005) proposed the following method of ELASTIC NET

$$\hat{\beta}_{\theta}^{\text{ENET}} = \text{argmin}_{\beta}\Big\{\|Y - X\beta\|^2 + \tau_1\|\beta\|^2 + \tau_2|\beta|\Big\},$$

where $\theta = (\tau_1, \tau_2)$ with nonnegative τ_1 and τ_2 is the tuning parameter vector. Compared with the ridge estimator $\hat{\beta}_{\lambda}^{\text{RIDGE}}$, the ELASTIC NET estimator $\hat{\beta}_{\theta}^{\text{ENET}}$ is able to do variable selection. Compared with the lasso estimator $\hat{\beta}_{\gamma}^{\text{LASSO}}$, the elastic net estimator $\hat{\beta}_{\theta}^{\text{ENET}}$ can deal with multi-collinearity and $p \gg n$ problems. For such a reason, the $\hat{\beta}_{\theta}^{\text{ENET}}$ has achieved quite a success in both theory and application.

However, for any given data one may have to try each of above methods individually to select the best estimate. In this paper, we propose a new method, generalized adaptive ridge (GAR), unifying adaptive RIDGE REGRESSION, ADAPTIVE LASSO and adaptive ELASTIC NET which is newly proposed in this paper. The paper is organized as the following: Section

2.2 describes the GAR method; Section 2.3 compares the results among the methods using simulations and real data. Section 2.4 presents a short discussion. The proof is given in the Appendix. For simplicity, the response variables are centered and the predictors are standardized.

2.2 Generalized adaptive ridge

We now propose our method: general adaptive ridge (GAR). Suppose $w_j \geq 0$ for $j = 1, \cdots, p$ are the weights and $W = \text{diag}(w_j)$, $\Lambda = \text{diag}\{\lambda_1, \cdots, \lambda_p\}$, with $\lambda_j > 0$ for $1 \leq j \leq p$, and $\lambda > 0$ is a predefined value. Under the constraint $\sum_{j=1}^{p} \frac{1}{\lambda_j} w_j = p$,

$$\hat{\beta}_{\lambda}^{\text{GAR}} = \arg\min_{\beta,\Lambda} \parallel \mathbf{Y} - X\beta \parallel^2 + \lambda \beta^{\top} \Lambda W \beta. \tag{2.2}$$

Grandvalet (1998) discussed an adaptive ridge regression which is a special case of GAR with $w_j = 1$ for $j = 1, \cdots, p$. To interpret the λ_j's, one may think that the Bayes prior distribution for β_j is a centered normal distribution with variance proportional to $1/(\lambda\lambda_j)$. The constraint is a link between the p prior distributions so that their weighted mean is proportional to $1/\lambda$, because $\frac{1}{p}\sum_{j=1}^{p} \frac{1}{\lambda\lambda_j} w_j = \frac{1}{\lambda}$.

Let $c = \sum_{j=1}^{p} \frac{1}{\lambda_0 + \lambda\lambda_j} w_j$, for $\lambda_0, \lambda \geq 0$,

$$\parallel \mathbf{Y} - X\beta \parallel^2 + \lambda_0 \beta^{\top} W \beta + \lambda \beta^{\top} \Lambda W \beta, \tag{2.3}$$

can be written by re-parameterization as $\parallel \mathbf{Y} - X\beta \parallel^2 + \lambda^* \beta^{\top} \Lambda^* W \beta$, where $\Lambda^* = \{\lambda_1^*, \cdots, \lambda_p^*\}$, $\lambda^* = \frac{p}{c}$ and $\lambda_j^* = c(\lambda_0 + \lambda\lambda_j)/p$. Then $\sum_{j=1}^{p} \frac{1}{\lambda_j^*} w_j = p$. Hence (2.3) is a special case of GAR in (2.2).

Shortly we will see that the solution of equation (2.3) is equivalent to a new method of adaptive ELASTIC NET which is described as the solution of β minimizing the following:

$$\|Y - X\beta\|^2 + \lambda_0 \beta^{\top} W \beta + \tau \sum_{j=1}^{p} w_j |\beta_j|, \tag{2.4}$$

where $\lambda_0, \tau \geq 0$ are predefined tuning parameters. The adaptive ELASTIC NET reduces to the ELASTIC NET (Zou and Hastie, 2005) if W is an identity matrix. Hence, it is a balance

between adaptive RIDGE REGRESSION and ADAPTIVE LASSO. Specifically, if $\tau = 0$, it is adaptive RIDGE REGRESSION; If $\lambda_0 = 0$, it is ADAPTIVE LASSO. If λ_0 and λ are chosen properly, this new penalization method will represent a balance in terms of power between adaptive RIDGE REGRESSION and ADAPTIVE LASSO. Hence ALTERNATIVE ELASTIC NET is an adaptive ELASTIC NET solution.

We now state the following main result whose proof is in the Appendix A1.

Proposition 1. *Suppose that w_j's for $j = 1, \cdots, p$ are fixed, then for a given $\lambda \geq 0$ there exists a $\tau \geq 0$ such that the solutions of the minimization problems of (2.3) and (2.4) are the same, and vice versa.*

Proposition 1 and the fact that (2.3) is a special case of (2.2) imply that adaptive ELASTIC NET is a special case of GAR. On the other hand, if $\lambda_0 = 0$, Proposition 1 shows that GAR is equivalent to ADAPTIVE LASSO. Hence, GAR can deal with correlated data or data with $p >> n$, and do variable selection as ADAPTIVE LASSO does. On the other hand, because of the ridge type formulation, GAR can be solved by simple iterative RIDGE REGRESSION algorithm. These features indicated that GAR is a unified method of all three. That is, if the regression parameters truly enjoy a sparse representation, GAR performs like the most recently proposed ADAPTIVE LASSO (Zou, 2006), hence, is able to identify relevant predictors consistently. If the regression parameters are not that sparse, GAR performs like the adaptive RIDGE REGRESSION, which is well-known for its high prediction accuracy and reliability against the multi-collinearity problem. If the predictor dimension is much larger than the sample size, and the parameters are sparse, GAR performs like adaptive ELASTIC NET. Therefore, it is a data-driven method.

For a given data $(\mathbf{Y}, X)$, and specified λ, Λ and W, GAR in (2.2) has a unified solution:

$$\beta = (X^\top X + \lambda \Lambda W)^{-1} X^\top \mathbf{Y}. \tag{2.5}$$

Hence, the important thing is how to choose proper W and λ, as well as how to find the optimized Λ satisfying the constraint $\sum_{j=1} \frac{1}{\lambda_j} w_j = p$ for our final solution, which will be dealt with in the next section.

2.2.1 Algorithm for gar

Adaptive weights: we follow the idea of Zou (2006) and use his suggestion of $W = \text{diag}\{|\hat{\beta}_j^{\text{OLS}}|^{-\nu}\}$ where $\nu \geq 0$ for $n > p$, and replace $\hat{\beta}^{\text{OLS}}$ by the ridge regression estimate with tiny tuning parameter for $n < p$. One may use trial and error to select ν as suggested by Zou (2006). In this paper we use $\nu = 0, 0.5, 1$ for the purpose of comparison. Using this way we set the initial weights ($W^{(0)}$) and update the weights dynamically instead of fixed weights in the iteration of our algorithm below.

Initial choice of parameters: we suggest MGCV (Wood, 2000) to estimate Λ initially by minimizing

$$V(\Lambda^{(0)}) = \frac{\|\mathbf{y} - A(\Lambda^{(0)})\mathbf{y}\|^2/n}{[Trace(I - A(\Lambda^{(0)}))/n]^2}, \tag{2.6}$$

where $A(\Lambda^{(0)}) = X(X^\top X + \Lambda^{(0)})^{-1} X^\top$ is called influence matrix for the model. This method is efficient in estimation of multiple tuning parameters corresponding to multiple penalizations. If $n > p$, we directly use above MGCV method; If $n < p$, we automatically enlarge the data by adding an identity $p \times p$ matrix to X and $p \times 1$ zero vector to $\mathbf{y}$ so that the artificial data has sample size bigger than p. This is allowed because we only use it to obtain initial Λ which will be updated in the algorithm anyway. Using Wood's MGCV algorithm by sending the proper data matrix and setting trivial parameters of 1's to obtain solutions $\hat{\Lambda}^{(0)}$ and $\hat{\beta}^{(0)}$. We then construct $\lambda^{(0)}\Lambda^{(0)} = \hat{\Lambda}^{(0)}\text{diag}\{(\hat{\beta}_j^{(0)})^{-2}\}$, where $\Lambda^{(0)}$ satisfies $\sum_{j=1}^{p} \frac{1}{\lambda_j^{(0)}} w_j^{(0)} = p$.

Based on the description of the iterative scheme in Appendix A2, we have suggested the following algorithm: For fixed $\lambda = \lambda^{(0)}$ and the accuracy $\delta > 0$ and $\delta_1 > 0$,

1. Set initials $p_0 = p$, $W^{(0)}$, $\Lambda^{(0)}$ and $s = 0$. Use (2.5) to obtain $\beta^{(s)}$.

2. For $j = 1, \cdots, p_s$, calculate $b_j^{(s)} = \frac{|\beta_j^{(s)}|}{\sum_{j=1}^{p_s} w_j^{(s)} |\beta_j^{(s)}|}$. If $b_j^{(s)} < \delta$, delete jth variable; if $b_j^{(s)} \geq \delta$, keep it.

3. Let p_{s+1} be the number of remain predictors in the model. For these variables, update $W^{(s+1)}$ by obtaining new OLS or RIDGE REGRESSION estimates, and update $\Lambda^{(s+1)}$ by $\lambda_j^{(s+1)} = (\frac{p|\beta_j^{(s)}|}{\sum_{j=1}^{p_{s+1}} w_j^{(s+1)} |\beta_j^{(s)}|})^{-1}$.

4. Update $\beta^{(s+1)}$ by equation (2.5) for the reduced variables, and calculate max $\frac{|\beta_j^{(s)} - \beta_j^{(s+1)}|}{1+|\beta_j^{(s+1)}|}$. If it is less than δ, stop; otherwise, go to step 2 and update until it stops.

We call this algorithm GAR.

Remark 1. In the algorithm, once variable is deleted, it never enters again. In this way, the computation becomes much more efficient. Also step 3 indicates that the constraint on Λ is satisfied automatically.

Remark 2. This algorithm is only a 'typical' ridge estimator, thus GAR provides an efficient and alternative approach for ELASTIC NET (when $W = I$). The convergence criterion is $\delta_1 = 10^{-6}$.

Remark 3. If $\nu = 0$, then $W = I$ throughout the algorithm. That is the same to set weights to be equal all the time in the algorithm, thus we refer it as GAR with equal weights. Our limited simulation later show that generally GAR with equal weights seems good enough for this method.

Our algorithm may be further improved by using the idea in Hawkins and Yin (2002), and Turlach (2006). However, the simulations and data analysis in section 2.3 seem show that the above algorithm is already effective enough. The R code for the algorithm is available from the authors.

2.2.2 Choice of parameters

In our algorithm, the accuracy δ is needed. Because as described in Appendix A2, a $\beta_k = 0$ will force $\frac{|\beta_k|}{\sum_{j=1}^{p} w_j |\beta_j|}$ to be zero in the iteration between β_k's and λ_k's. Therefore, practically,

a small $\frac{|\beta_k|}{\sum_{j=1}^{p} w_j|\beta_j|}$ may indicates the respective $\beta_k = 0$. Hence we need a threshold $\delta > 0$. For fixed tuning parameter λ, the results will be affected by the accuracy δ.

Unlike other algorithms, in our case, we set $0 < \delta \leq 1$. The tuning parameter $\lambda \in (0, \Delta]$, where Δ is an integer ($\Delta = 100$ in our simulations). We select a series of λ and a series of δ in the respective intervals. Our algorithm finds the solution for each pair of (λ, δ) in a matrix. And finally, we select the pair of (λ, δ) with the smallest of either AIC (Akaike, 1973), BIC (Schwarz, 1978) or RIC (Shi and Tsai, 2002):

$$\begin{aligned}
\mathrm{AIC}_\delta(\lambda) &= n\log(\|Y - X\beta_\delta(\lambda)\|^2/n) + 2p_\delta(\lambda) \\
\mathrm{BIC}_\delta(\lambda) &= n\log(\|Y - X\beta_\delta(\lambda)\|^2/n) + \log(n)p_\delta(\lambda) \\
\mathrm{RIC}_\delta(\lambda) &= (n - p_\delta(\lambda))\log(\|Y - X\beta_\delta(\lambda)\|^2/(n - p_\delta(\lambda))) + p_\delta(\lambda)(\log(n) - 1) + 4/(n - p_\delta(\lambda) - 2)
\end{aligned}$$

We first note that from our limited simulations in selecting tuning parameters, AIC, BIC and RIC give very similar results when $n > p$, and AIC and BIC give very similar results when $n < p$, although the respective curves of the parameters may be different (Plot of criterion vs. λ marked by δ such as in Figures 2.1 and 2.2). Thus we only report the BIC criterion in this paper.

We find that when $n > p$ across λ, there is a certain range of δ which is the same for the corresponding BIC criterion such that the values of them in each column of λ are among the smallest and stable, respectively. For instance, in Figure 2.1 for Model 1 in section 2.3.1, for all λ's, the BIC curves are the same for $\delta = .01, ..., 1$. These selected δ will produce the same result. On the other hand, across δ, there is a certain range of λ which is the same for the corresponding BIC criterion such that the values of them in each row of δ are among the smallest and stable, respectively. For example, in Figure 2.1 BIC values are the smallest and stable for $0 < \lambda < .5$. One may choose any value for λ in this range. Within the range, again different values produce the same results. However, in our limited simulations and data analysis, the $\lambda = \lambda^{(0)}$ that is selected by MGCV method in the previous section, is always in this range of λ. For model 1, the vertical line (close to .05) is the MGCV for λ. Thus we use $\lambda = \lambda^{(0)}$ as the selected tunning parameter. When $n < p$, the selected λ by MGCV may

not be the best as in Figure 2.2 where the MGCV λ is the vertical line which is closed to 4. The best values of λ is in between 0 and .2. However, the results by using MGCV λ and the best λ have little difference.

The tuning parameter selection for Model 3 has a similar figure to the Figure 2.1 of model 1, while the figure for Model 4 is very similar to Figure 2.2 of model 2. We find that our selection method for tuning parameters is very useful and efficient in the simulations.

2.3 Numerical studies

In this section, we shall compare the results among GAR, ADAPTIVE LASSO, LASSO, RIDGE REGRESSION and ELASTIC NET via a small simulation study as well as one data set. All ELASTIC NET and LASSO are performed in R package, using 10-fold cross-validation method to select the best tuning parameters. And ADAPTIVE LASSO is performed by using Zou's code (through personal communication).

2.3.1 Simulations

In this simulation part, we consider our algorithm with weights. The weight vector was set as $\hat{w} = 1/|\hat{\beta}|^{\nu}$ (Zou, 2006), where $\hat{\beta}$ is OLS estimate for $(n > p)$ and $\hat{\beta}$ is ridge estimate for $(n <= p)$. For comparison purpose we fixed $\nu = 0, .5, 1$ as did in Zou (2006). When $\nu = 0$, this is our special case of equal weight. Four models are considered.

Model 1. This model is called inconsistent LASSO path (Zou, 2006), which was used to make comparison with ADAPTIVE LASSO developed by Zou (2006). The model is defined as $y = X^{\top}\beta + N(0, \sigma^2)$, where $\beta = (5.6, 5.6, 5.6, 0)$, X_i, $i = 1, ..., n$ are iid $N(0, C)$ and C is a 4×4 symmetric matrix. The 3×3 block matrix $C_{11} = (1 - \rho_1)I + \rho_1 J$, where I is the 3×3 identity matrix, J is the 3×3 matrix of 1's and $\rho_1 = -.39$. The 3×1 block vectors are, $C_{21}^{\top} = C_{12} = \rho_2 \overrightarrow{1}$, where $\overrightarrow{1}$ is the 3×1 vector of 1's and $\rho_2 = .23$. Finally, $C_{22} = 1$.

Table 2.1 presents a simulation with combinations of n, σ and ν using the way in Zou (2006). We generated 100 datasets for each set up. We obtained solution by the respective

method for each dataset, then reported the percentage of the 100 solutions selecting the true model. RIDGE REGRESSION is not performed since it selects all the variables. Since GAR is equivalent to ADAPTIVE LASSO, as expected, GAR can perform a consistent variable selection. In fact its results are comparable with ADAPTIVE LASSO when $n = 60$, and better when $n = 120, 300$. Note that weights seem not so critical for the results, which may be due to the dynamic update of the weights in our algorithm. Confirming with Zou's report, LASSO performs worse than that of ADAPTIVE LASSO, and also in this set up, ELASTIC NET is expected to perform as LASSO does. In detailed results, we find that ELASTIC NET and LASSO often overfit the model.

For the next three comparisons, we consider the following model:

$$Y = \beta^\top \mathbf{X} + \sigma_\epsilon \epsilon,$$

where $X \sim N_p(0, \Sigma)$, $\epsilon \sim N(0, 1)$, β is $p \times 1$ vector and n is the sample size.

Model 2. All elements of β are .86's, and $\Sigma = I$, $\sigma_\epsilon = .5$, $n = 50$, $p = 100$. We replicate 100 data sets. For this model, RIDGE REGRESSION should be the best method.

Table 2.2 reports the means and standard errors for the absolute difference $|\beta - \hat{\beta}|$, the correlation coefficient between $\beta^\top \mathbf{X}$ and $\hat{\beta}^\top \mathbf{X}$ for different methods. Although RIDGE REGRESSION seems outperform GAR a little, both difference and correlation showed that the results are almost equivalent. Which means that GAR doesn't lose power to RIDGE REGRESSION even in case that RIDGE REGRESSION is the best method for the model. With increase of weight ν, absolute difference $|\beta - \hat{\beta}|$ for GAR increases. But the difference seems not so big. And the respective correlation coefficients seems unchanged. In this case, GAR seems not very sensitive to weight. On the other hand, GAR again outperforms ELASTIC NET. LASSO and ADAPTIVE LASSO are not proper methods, due to the set-up of $n < p$.

In the next two models, we report the way similar to that of Wang, Li and Tsai (2007). We define $S_T = \{j : \beta_j \neq 0\}$, the true model, and $S_e = \{j : \hat{\beta}_j \neq 0\}$, the estimated model. Thus the sets of $\{e : S_e \not\supset S_T\}$, $\{e : S_e = S_T\}$ and $\{e : S_e \supset S_T \text{ and } S_e \neq S_T\}$, are the underfitted,

correctly fitted and overfitted models, respectively. We report the rate of these models over the total number of models. Furthermore, in the overfitted case, we also reported the rate of how many parameters are overfitted. The average of 0 coefficients is also reported, in which the column labeled "C" presents the average restricted only to the true zero coefficients, and the column labeled "I" depicts the average of coefficients erroneously set to 0. The model error is defined as

$$\mathrm{ME}(\hat{\beta}) = E(x^\top \hat{\beta} - x^\top \beta)^2 = (\hat{\beta} - \beta)^\top E(x^\top x)(\hat{\beta} - \beta),$$

and the relative model error is defined as $\mathrm{ME}(\hat{\beta})/\mathrm{ME}(\hat{\beta}_o)$ where $\hat{\beta}_0$ is the usual OLS fit for model 3, and $\mathrm{ME}(\hat{\beta})/\mathrm{ME}(\hat{\beta}_r)$ where $\hat{\beta}_r$ is the RIDGE REGRESSION fit (using Wood's ridge code with default tuning parameter) for model 4, respectively. We report the median of the relative model error (MRME) over 1000 simulated datasets.

Model 3. This model is originally used by Tibshirani (1996), with $p = 8$, $\beta^\top = (3, 1.5, 0, 0, 2, 0, 0, 0)$. But each element of Σ is $\rho^{|i-j|}$ for all i and j and $\sigma_\epsilon = 1$, with $n = 200$, and $\rho = 0.25$ and 0.5. LASSO should be the best method in terms of variable selection. Table 2.3 shows that all the methods are equally good in terms of variables selection in that they all correctly select the models all the time. However, in terms of prediction errors via MRME, GAR is the best, ADAPTIVE LASSO is closely the second best. Because their respective MRMEs are close. Equal weight of GAR seems work well again. As expected, LASSO and ELASTIC NET performed similarly. But their prediction errors are much bigger. The result of LASSO seems different from what Fan and Li (2001) reported. However, in their report LASSO overfitted model, thus has smaller MRME. We also find that the MRME reduces to what they had, if LASSO overfitted the model in our study. But with sample size of $n = 200$, LASSO correctly fits the model.

Model 4. The same model as model 3, except $p = 100$ and $n = 50$ with additional $\rho = .75$. While the first eight elements of β is $(3, 1.5, 0, 0, 2, 0, 0, 0)$, the rest are zeros. ELASTIC NET should be the best method. We don't perform RIDGE REGRESSION because

it doesn't select variables. GAR with equal weights seems the best method here, the differences among using weights seem not that significant. GAR identifies more correct zeros than ELASTIC NET does. Although GAR occasionally underfits the model, its MRME is significantly smaller than that of ELASTIC NET. On the other hand, ELASTIC NET seems overfit the model more often. GAR consistently outperforms that of ELASTIC NET in this model.

Note that in our comparisons of these models, it appears there is a 'clear winner' for each model, and not all the methods are suitable for each model. In fact, GAR and ELASTIC NET are the only methods that are suitable for across all the models among these methods. However, in all suitable comparisons, GAR is very comparable with the 'winner' method in each case: in model 1 with ADAPTIVE LASSO but better than LASSO and ELASTIC NET, and in model 2 with RIDGE REGRESSION but better than ELASTIC NET; or better than the 'winners' LASSO and ADAPTIVE LASSO in model 3 and ELASTIC NET in model 4. The limited simulations seem suggest that regardless of the data type, GAR works well and also GAR with equal weights does a great job.

2.3.2 A DIABETES DATA

To demonstrate the usefulness of the GAR method on real dataset, we revisit the famous diabetes study data (Efron, Hastie, Johnstone and Tibshirani 2004). The predictors are age, sex, body mass index (bmi), mean arterial pressure (map), and six serum measurements (tc, ldl, hdl, tch, ltg, glu). The response variable is a quantitative measure of disease progression. There are 442 diabetes patients involved in this study.

We randomly select half of the data to select the tuning parameters and then use the other half to select the model. We do this 100 times, and Table 2.5 reports the results. Weights again have little effect for GAR which agrees with the performance in our simulations. It seems that ELASTIC NET and LASSO tend to select variables similarly, which agrees with Efron et al.'s (2004) report for LASSO. On the other hand, ADAPTIVE LASSO and GAR tend to agree more than that of GAR and ELASTIC NET, supporting the conclusion from simulations in

models 1 and 3. We don't know the true model in the dataset. However, this dataset may be a type of model 1 or model 3 or a combination of these twos, since $n > p$. In such a case, ADAPTIVE LASSO and GAR may have better ability to select variables. On the other hand, we may look at the prediction errors. Table 2.6 reports the comparison of the performance of those methods in terms of the two average mean squared errors (MSE). The in-sample MSE is the MSE for the sample that is used to fit the model, the out-sample is the MSE for the other sample. For in-sample MSE, GAR seems a little bit smaller than that of all others which again agrees what we conclude in model 3. For those of out-sample MSEs, GAR with weights ($\nu = .5, 1$) have the two smallest prediction errors. The three methods of LASSO, GAR with equal weights, and ELASTIC NET have produced similar errors. However, LASSO and ELASTIC NET did select more variables. The two ADAPTIVE LASSO also have relative big errors. Overall if the dataset is indeed a combination of models 1 and 3, then the conclusion from variable selection and prediction error is that GAR may be the best method. Because GAR selects the smallest number of variables with smallest prediction errors.

We further looked at details for the data. First we randomly select half of the data to choose the tuning parameters. Then we use the whole data to select important variables. GAR with equal weights selects three variables (NO. 3, 4 and 9: bmi, map and ltg, respectively); GAR with weights ((ν) = 0.5 and 1) selects two variables (NO. 3 and 9: bmi and ltg, respectively); ELASTIC NET selects four variables (NO. 3, 4, 7, 9: bmi,map,hdl, ltg, respectively); LASSO selects four variables (NO. 3, 4, 7, 9: bmi,map,hdl, ltg, respectively); Adaptive LASSO with weight equal 0.5 selects four variables (NO. 3, 4, 7, 9: bmi,map,hdl, ltg, respectively); Adaptive LASSO with weight equal 1 selects four variables (NO. 3, 4, 5, 9: bmi,map,tc, ltg, respectively); These results support Table 2.5.

2.4 Discussion

In this article, we showed that in theory data-driven GAR is a flexible and automatic method for (adaptive-) RIDGE REGRESSION, LASSO and ELASTIC NET. Our limited simulations show

that the adaptiveness in GAR seems not so important, thus equal weights in GAR is efficient. This may be due to the algorithm that dynamically updates weights in each step. In comparing with RIDGE REGRESSION, LASSO and ELASTIC NET, our performance is essentially equivalent or better. The overall performance of numerical efficiency of GAR may be due to its flexibility in the algorithm.

Acknowledgement

We would like to thank Editor's constructive comments and Professor Hui Zou for providing us his ADAPTIVE LASSO algorithm.

APPENDIX: JUSTIFICATION

A1. Proposition 2. Define $\gamma_j = \sqrt{\lambda_j}\beta_j; c_j = \sqrt{\frac{1}{\lambda_j}}$, for $j = 1, \cdots, p$, then $\gamma_j c_j = \beta_j$ and $\sum_{j=1}^p w_j c_j^2 = p$. Equation (2.3) then becomes

$$\hat{\beta}_\lambda^{\text{GAR}} = \arg\min \sum_{i=1}^n (y_i - \sum_{j=1}^p \gamma_j c_j x_{ij})^2 + \lambda_0 \sum_{j=1}^p w_j \gamma_j^2 c_j^2 + \lambda \sum_{j=1}^p w_j \gamma_j^2, \text{ under } \sum_{j=1}^p w_j c_j^2 = p.$$

To solve this minimization problem, we have the Lagrangian

$$\hat{\beta}_\lambda^{\text{GAR}} = \arg\min \sum_{i=1}^n (y_i - \sum_{j=1}^p \gamma_j c_j x_{ij})^2 + \lambda_0 \sum_{j=1}^p w_j \gamma_j^2 c_j^2 + \lambda \sum_{j=1}^p w_j \gamma_j^2 + m(\sum_{j=1}^p w_j c_j^2 - p).$$

By taking derivatives with respect to c_k and γ_k , we have three equations:

$$\gamma_k T_k + \lambda_0 w_k \gamma_k^2 c_k = -m w_k c_k, c_k T_k + \lambda_0 w_k \gamma_k c_k^2 = -\lambda w_k \gamma_k, \text{ and } \sum_{j=1}^p w_j c_j^2 = p,$$

where $T_k = \sum_{i=1}^n x_{ik}(\sum_{j=1}^p \gamma_j c_j x_{ij} - y_i) = \sum_{i=1}^n x_{ik}(\sum_{j=1}^p \beta_j x_{ij} - y_i)$. We have two cases:

Case (1), $\gamma_k = 0$ leading to $\beta_k = 0$;

Case (2), $\gamma_k \neq 0$, multiple the first equation by c_k and second one by γ_k, we have $m w_k c_k^2 = \lambda w_k \gamma_k^2$, which by the way defined for for γ_k and c_k leads to

$$\frac{\gamma_k}{c_k} = \sqrt{\frac{m}{\lambda}} sign(\beta_k) \tag{2.7}$$

Divide the first two equations by $w_k \gamma_k$ and $w_k c_k$, respectively, we have

$$\frac{T_k}{w_k} + \lambda_0 \beta_k = -m\frac{c_k}{\gamma_k}; \frac{T_k}{w_k} + \lambda_0 \beta_k = -\lambda \frac{\gamma_k}{c_k}. \tag{2.8}$$

Multiple the two equations, we have $(\frac{T_k}{w_k} + \lambda_0\beta_k)^2 = \lambda m$, which is a constant and denoted by T_0^2 where $T_0 = \frac{T_k}{w_k} + \lambda_0\beta_k$. From the first equation , we have

$$\gamma_k T_0 = -mc_k, w_k c_k \gamma_k T_0 = -m w_k c_k^2, w_k \beta_k T_0 = -m w_k c_k^2, w_k|\beta_k||T_0| = m w_k c_k^2,$$

using the fact that $\sum_{j=1}^p w_j c_j^2 = p$, we have $|T_0| \sum_{j=1}^p w_j|\beta_j| = mp$. And similarly, from the second equation, we have $|T_0|p = \lambda \sum_{j=1}^p w_j|\beta_j|$. The ratio of these two equations leads to $m = \frac{\lambda(\sum_{j=1}^p w_j|\beta_j|)^2}{p^2}$. Put this and (2.7) into the second equation in (2.8), we have $T_k + \lambda_0 w_k\beta_k + w_k\frac{\lambda}{p} sign(\beta_k) \sum_{j=1}^p w_j|\beta_j| = 0$. That is, the solutions of case (2) satisfies

$$\sum_{i=1}^n x_{ik}(\sum_{j=1}^p \beta_j x_{ij} - y_i) + \lambda_0 w_k\beta_k + w_k\frac{\lambda}{p} sign(\beta_k) \sum_{j=1}^p w_j|\beta_j| = 0. \tag{2.9}$$

However, $\beta_k = 0$ and (3.14) are the the solution of the following minimization problem:

$$\hat{\beta} = \arg\min \sum_{i=1}^n (y_i - \sum_{j=1}^p \beta_j x_{ij})^2 + \lambda_0 \beta^\top W \beta + \frac{\lambda}{p}(\sum_{j=1}^p w_j|\beta_j|)^2,$$

which is equivalent to $\hat{\beta} = \arg\min \sum_{i=1}^n (y_i - \sum_{j=1}^p \beta_j x_{ij})^2 + \lambda_0\beta^\top W\beta$, under $\frac{\lambda}{p}(\sum_{j=1}^p w_j|\beta_j|)^2 < t'$; which is then equivalent to

$$\hat{\beta} = \arg\min \sum_{i=1}^n (y_i - \sum_{j=1}^p \beta_j x_{ij})^2 + \lambda_0 \beta^\top W \beta \tag{2.10}$$

under $\sum_{j=1}^p w_j|\beta_j| < t$ for $t = \sqrt{pt'/\lambda}$. But (2.10) is equivalent to (2.4). Hence the proof is complete. □.

A2. Algorithm background. Using the notation in the proof of Proposition 2, and define $D_c = \text{diag}\{c_1, \cdots, c_p\}$ and $\boldsymbol{\gamma}^\top = (\gamma_1, ..., \gamma_p)$. By this re-parameterization, the solution can be obtained from D_c and $\boldsymbol{\gamma}$. Thus for giving λ and W, if D_c is given, then

$$\boldsymbol{\gamma} = (D_c X^\top X D_c + \lambda W)^{-1} D_c X^\top Y. \tag{2.11}$$

If $\boldsymbol{\gamma}$ is given, from the proof, $c_k^2 = \gamma_k^2\frac{\lambda}{m} = \frac{\gamma_k^2 p^2}{(\sum w_j|\beta_j|)^2}$, by the constraint $\sum w_j c_j^2 = p$, we have

$$c_k^2 = \frac{p\gamma_k^2}{\sum w_j \gamma_j^2} \tag{2.12}$$

Thus an alternating algorithm can be developed by using (2.11) and (2.12).

However, one can easily see that (2.11) is the same as (2.5). And furthermore, $c_k^2 = \frac{\gamma_k^2 p^2}{(\sum w_j|\beta_j|)^2}$ implies that $c_k^4 = \frac{\gamma_k^2 c_k^2 p^2}{(\sum w_j|\beta_j|)^2} = \frac{\beta_k^2 p^2}{(\sum w_j|\beta_j|)^2}$. Hence $c_k^2 = \frac{p|\beta_k|}{\sum w_j|\beta_j|}$. Thus

$$\lambda_k = \frac{1}{c_k^2} = (\frac{p|\beta_k|}{\sum w_j|\beta_j|})^{-1}, \tag{2.13}$$

where the constraint on Λ is automatically satisfied. Therefore, an equivalent algorithm can be developed iteratively between (2.5) and (3.15). That is, for given λ_k's, find β by (2.5); check c_k^2 or $c_k^2/p = \frac{|\beta_k|}{\sum w_j|\beta_j|}$ if it is smaller than a threshold, delete the respective β_k; otherwise keep it there. Then for the remain variables, update the respective λ_k by (3.15). Proceed this scheme iteratively until it is convergent. □.

2.5 References

[1] Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. *Second International Symposium on Information Theory*, eds. Petrov, B.N., and Csaki, F., 261-281.

[2] Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression. *Ann. Statist.* 32, 407499.

[3] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* 96, 1348–1360.

[4] Grandvalet, Y. (1998). Least absolute shrinkage is equivalent to quadratic penalization. In *L. Niklasson, M. Boden, and T. Ziemske, editors, ICANN98* pages 201-206. Springer.

[5] Hawkins, D. and Yin, X. (2002). A faster algorithm for ridge regression for reduced rank data. *Journal of Computational Statistics & Data Analysis* 40, 253–262.

[6] Hoerl, A. E. and Kennard, R. W. (1970a). Ridge regression: applications to nonorthogonal problems. *Technometrics* 12, 69–82.

[7] Hoerl, A. E. and Kennard, R. W. (1970b). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* 12, 55–67.

[8] Leng, C., Lin, Y. and Wahba, G. (2006). A note on lasso and related procedures in model selection. *Statistica Sinica* 16, 1273–1284.

[9] Schwarz, G. (1979). Estimating the dimension of a model. *Annals of Statistics*, 6, 461-464.

[10] Shi, P., and Tsai, C.-L. (2002). Regression model selection – a residual likelihood approach. *Journal of Royal Statistical Society, Series B.* 64, 237-252.

[11] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B* 58, 267-288.

[12] Turlach, B. A. (2006). An even faster algorithm for ridge regression of reduced rank data. *Journal of Computational Statistics & Data Analysis*, 50, 642–658.

[13] Wang, H., Li, R. and Tsai, C .L. (2007). Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika* 94, 553–568.

[14] Wang, H., Li, G. and Jiang, G. (2007). Robust regression shrinkage and consistent variable selection via the lad-lasso. *Journal of Business and Economics Statistics* 25, 347–355.

[15] Wang, H., Li, G. and Tsai, C. L. (2007). Regression coefficient and autoregressive order shrinkage and selection via lasso. *Journal of Royal Statistical Society, Series B* 69, 63–78.

[16] Wood, S. N. (2000). Modeling and Smoothing Parameter Estimation with Multiple Quadratic Penalties. *J. R. Statist. Soc. B* 62, 413-428

[17] Yuan, M. and Lin, Y. (2007). On the nonnegative garrote estimator. *Journal of the Royal Statistical Society, Series B* 69, To appear.

[18] Zhang, H. H. and Lu, W. (2007). Adaptive lasso for cox's proportional hazard model. *Biometrika*, To appear.

[19] Zhao, P. and Yu, B. (2007). On model selection consistency of lasso. *Journal of Machine Learning Research* 7, 2541–2567.

[20] Zou, H. (2006). The adaptive-LASSO and its oracle properties. *Journal of American Statistical Association* 101, 1418-1429.

[21] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. R. Stat. Soc. B* 67, 301-320.

[22] Zou, H., Hastie, T. and Tibshirani, R. (2007). On the 'degrees of freedom' of lasso. *The Annals of Statistics*, 35, 2173-2192.

Table 2.1: Model 1: comparisons among ELASTIC NET, LASSO, adaptive LASSO and GAR: the percentage of selecting the true model

Method	$n = 60, \sigma = 9$	$n = 120, \sigma = 5$	$n = 300, \sigma = 3$
ELASTIC NET	.49	.50	.48
LASSO	.46	.50	.38
adaptive LASSO($\nu = .5$)	.56	.65	.90
adaptive LASSO($\nu = 1$)	.67	.89	1
GAR($\nu = 0$)	.54	.93	.97
GAR($\nu = .5$)	.55	.95	1
GAR($\nu = 1$)	.59	.97	1

The tuning parameters for GAR are MGCV λ and $\delta = .1$ as in Figure 1.

Table 2.2: Model 2: comparison among RIDGE REGRESSION, ELASTIC NET and GAR

	Mean absolute difference (se)			Mean correlation coefficients (se)		
Pairs	β VS β_{R}	β VS β_{GAR}	β_{GAR} VS β_{R}	β vs. β_{R}	β vs β_{GAR}	β_{GAR} vs β_{R}
$\nu = 0$	.5045(.3436)	.5404(.3479)	.0881(.0809)	.996(.0011)	.997(.0011)	.999(.0005)
$\nu = 0.5$	.5045(.3436)	.5414(.3421)	.0934(.0921)	.996(.0011)	.997(.0013)	.999(.0007)
$\nu = 1$	.5045(.3436)	.5423(.3451)	.1171(.1062)	.996(.0011)	.997(.0015)	.999(.0007)
Pairs	β VS β_{ET}	β_{ET} VS β_{R}	β_{GAR} VS β_{ET}	β vs. β_{ET}	β_{ET} vs β_{R}	β_{GAR} vs β_{ET}
$\nu = 0$	.6045(.3458)	.5828(.3396)	.2482(.2626)	.995(.0015)	.994(.0016)	.998(.0011)
$\nu = 0.5$	.6045(.3458)	.5828(.3396)	.2461(.2630)	.995(.0015)	.994(.0016)	.999(.0012)
$\nu = 1$	.6045(.3458)	.5828(.3396)	.2583(.3012)	.995(.0015)	.994(.0016)	.999(.0012)

β_{R} is the RIDGE REGRESSION estimate, β_{GAR} is the GAR estimate and β_{ET} is the ELASTIC NET estimate. The tuning parameters for GAR are $\lambda = .1$ and $\delta = .1$ as in Figure 2. While for RIDGE REGRESSION, the tuning parameter is selected by the typical GCV method.

Table 2.3: Model 3: Simulation results for n>p

Method	σ_ϵ	n	ρ	under fitted	correctly fitted	overfitted			Avg. No. of zeros		MRME
						1	2	≥ 3	I	C	
GAR($\nu = 0$)	1	200	0.25	0	1	0	0	0	0	5	0.3521
GAR($\nu = 0.5$)	1	200	0.25	0	1	0	0	0	0	5	0.3422
GAR($\nu = 1$)	1	200	0.25	0	1	0	0	0	0	5	0.3499
LASSO	1	200	0.25	0	1	0	0	0	0	5	1.3844
Adaptive LASSO($\nu = 0.5$)	1	200	0.25	0	1	0	0	0	0	5	0.4167
Adaptive LASSO($\nu = 1$)	1	200	0.25	0	1	0	0	0	0	5	0.3551
ELASTIC NET	1	200	0.25	0	1	0	0	0	0	5	1.3757
GAR($\nu = 0$)	1	200	0.50	0	1	0	0	0	0	5	0.3213
GAR($\nu = 0.5$)	1	200	0.50	0	1	0	0	0	0	5	0.3500
GAR($\nu = 1$)	1	200	0.50	0	1	0	0	0	0	5	0.3240
LASSO	1	200	0.50	0	1	0	0	0	0	5	1.3749
Adaptive LASSO($\nu = 0.5$)	1	200	0.50	0	1	0	0	0	0	5	0.4044
Adaptive LASSO($\nu = 1$)	1	200	0.50	0	1	0	0	0	0	5	0.3603
ELASTIC NET	1	200	0.50	0	1	0	0	0	0	5	1.3698

Table 2.4: Model 4: Simulation results for n<p

Method	σ_ϵ	n	ρ	under fitted	correctly fitted	overfitted			Avg. No. of zeros		MRME
						1	2	≥ 3	I	C	
GAR($\nu = 0$)	1	50	0.25	0.045	0.954	0.001	0.00	0.00	0.045	96.99	0.05087
GAR($\nu = 0.5$)	1	50	0.25	0.01	0.96	0.03	0.00	0.00	0.01	96.97	0.05001
GAR($\nu = 1$)	1	50	0.25	0.00	0.90	0.01	0.09	0	0.00	96.89	0.06146
ELASTIC NET	1	50	0.25	0	0	0	0	1	0	94	0.33570
GAR($\nu = 0$)	1	50	0.50	0.009	0.990	0.001	0	0	0.009	96.98	0.04967
GAR($\nu = 0.5$)	1	50	0.50	0.01	0.93	0.06	0	0	0.01	96.94	0.04890
GAR($\nu = 1$)	1	50	0.50	0.00	0.95	0.04	0.01	0.00	0.00	96.94	0.04415
ELASTIC NET	1	50	0.50	0	0	0	0.001	0.999	0	94.001	0.25344
GAR($\nu = 0$)	1	50	0.75	0.01	0.96	0.03	0	0	0.01	96.97	0.04501
GAR($\nu = 0.5$)	1	50	0.75	0.01	0.96	0.03	0.00	0	0.01	96.93	0.04321
GAR($\nu = 1$)	1	50	0.75	0.02	0.94	0.04	0	0	0.02	96.93	0.04239
ELASTIC NET	1	50	0.75	0	0	0.009	0.002	0.989	0	94.02	0.18091

Table 2.5: Diabetes data: percentage of variable selected over 100 runs by ELASTIC NET and GAR

	Var1	Var2	Var3	Var4	Var5	Var6	Var7	Var8	Var9	Var10
ELASTIC NET	0.01	0.46	1.00	1.00	0.00	0.02	0.97	0.21	1.00	0.33
LASSO	0.01	0.51	1.00	1.00	0.02	0.04	0.99	0.06	1	0.37
Adaptive LASSO($\nu = 0.5$)	0.00	0.09	1.00	0.99	0.46	0.02	0.43	0.01	1.00	0.00
Adaptive LASSO($\nu = 1$)	0.00	0.03	1.00	1.00	0.81	0.00	0.12	0.04	1.00	0.00
GAR($\nu = 0$)	0.00	0.00	0.95	0.82	0.13	0.11	0.40	0.09	0.91	0.02
GAR($\nu = 0.5$)	0.00	0.00	0.94	0.80	0.12	0.09	0.38	0.08	0.89	0.01
GAR($\nu = 1$)	0.00	0.00	0.93	0.79	0.10	0.08	0.37	0.06	0.88	0.01

Table 2.6: Diabetes data: comparison between ELASTIC NET and GAR

	MSE in-sample	MSE out-sample
ELASTIC NET	3025.62(196.53)	3124.50(203.34)
LASSO	3017.97(214.92)	3141.46(181.62)
Adaptive LASSO($\nu = 0.5$)	3072.38(169.06)	3165.31(170.70)
Adaptive LASSO($\nu = 1$)	3018.75(181.98)	3160.19(189.08)
GAR($\nu = 0$)	2954.62(236.72)	3132.85(257.07)
GAR($\nu = 0.5$)	2796.35(167.00)	3057.86(192.57)
GAR($\nu = 1$)	2802.55(164.84)	3059.02(189.82)

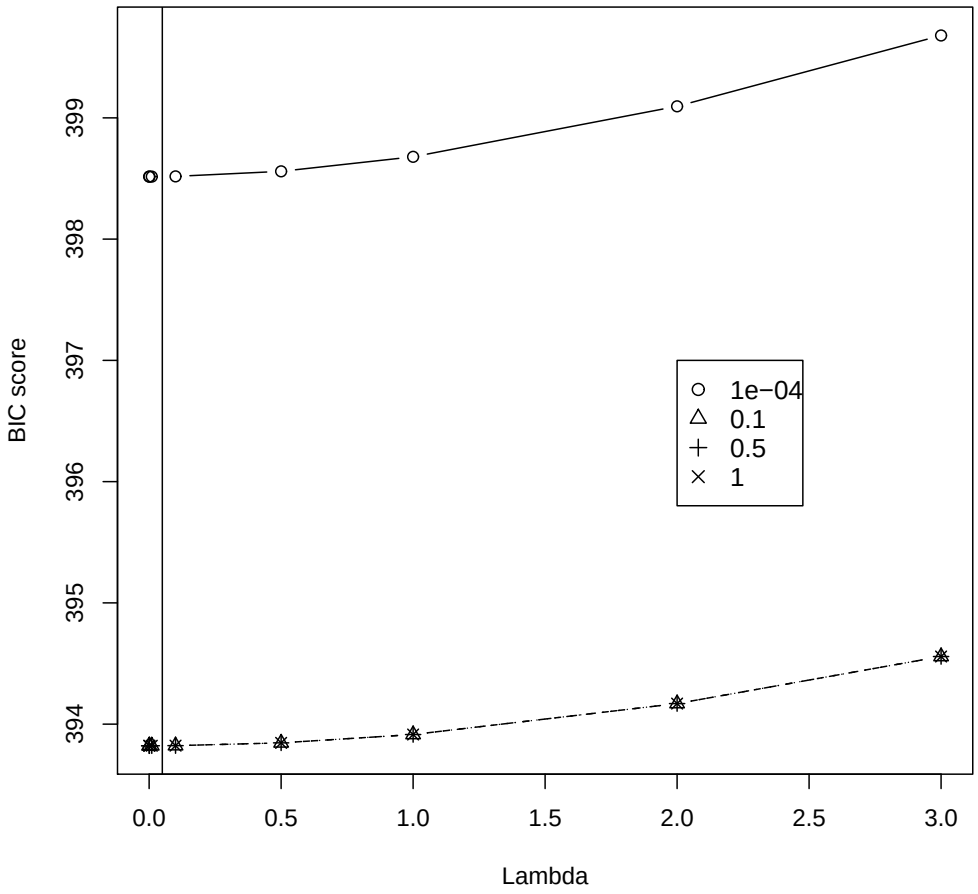

Figure 2.1: Tuning parameters selection (BIC) for Model I; The legend is for parameter δ.

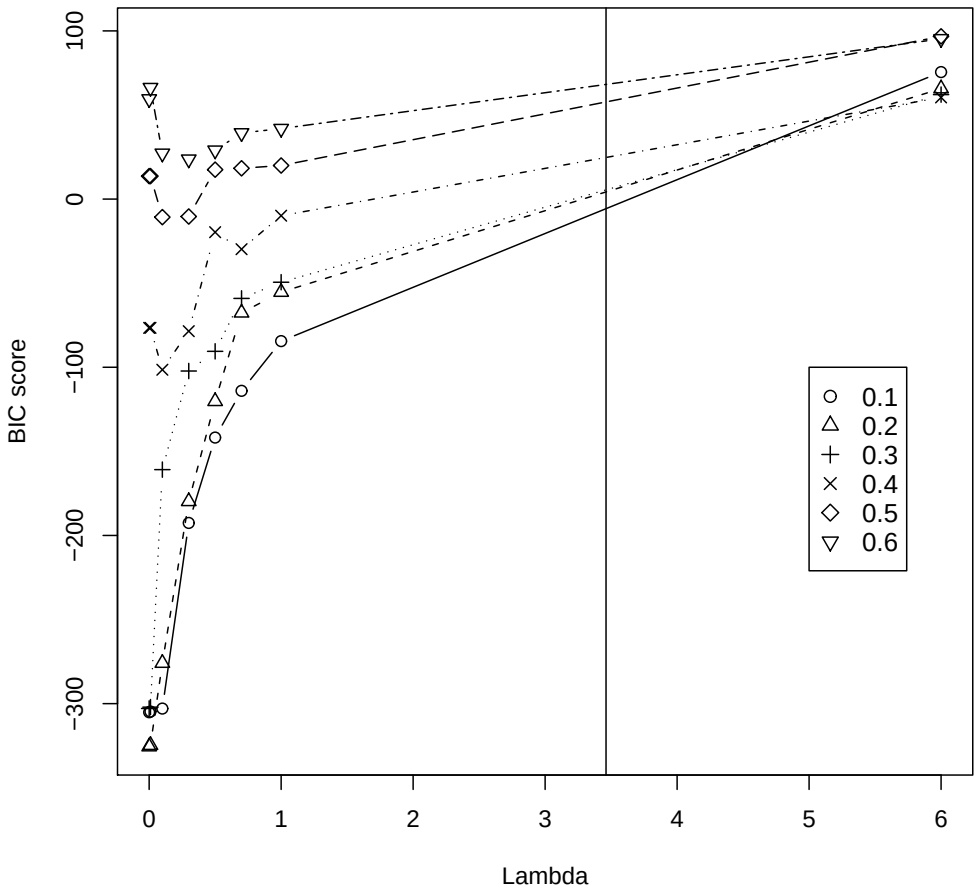

Figure 2.2: Tuning parameters selection for Model II; The legend is for parameter δ.

Chapter 3

A general adaptive L^2-regularization method[1]

[1]Qiu, J., Yin, X. Submitted to *Biometrics,* 6/2008.

Abstract

We consider a general adaptive L^2-regularized optimization problem $\hat{\beta}(\lambda) = \arg\min_{\beta,\Lambda} \ell(y,\beta) + \lambda\beta^T\Lambda W\beta$, where ℓ is a loss function, Λ and W are two diagonal matrices. We show that with appropriate choice of λ and Λ, if ℓ is differentiable, then the above adaptive L^2 penalty term is equivalent to adaptive L^1 penalty, adaptive L^2 penalty, and combined adaptive L^1 and L^2 penalty. Therefore, this method is a data-driven method, which automatically choose a penalty among the three penalty terms. We consider details when ℓ is a negative log-likelihood function which covers generalized linear model, and develop two algorithms using Newton-Raphson method for the general approach, and sequential minimal optimization (SMO) method for the case $p >> n$. The efficacy of our approach is illustrated by simulations, comparisons with other methods and real data analysis.

Key Words: Generalized linear model; Penalized likelihood; Variable selection; Penalization

3.1 Introduction

A general regularization optimization problem can be written as

$$\hat{\beta}(\lambda) = \arg\min_{\beta} \ell(y,\beta) + J_\lambda(\beta), \tag{3.1}$$

where ℓ is a loss function and J is a penalty term with respect to the given data $\{(x_i, y_i) : x_i \in R^p, y_i \in R, i = 1, \cdots, n\}$. For squared error loss, the most well-known penalized methods are RIDGE REGRESSION regression (Hoerl and Kennard, 1970a, b) which can deal correlated data and the case of $n << p$; the LASSO (Tibshirani, 1996) for variable selection; and the ELASTIC NET (Zou and Hastie, 2005) for simultaneously dealing with correlated data or $n << p$ and variable selection. Zou (2006) proposed Adaptive LASSO for squared loss to improve the LASSO. Both LASSO and adaptive LASSO have been used in other loss functions, for instance Cox's model (Tibshirani, 1997; Zhang and Lu, 2007), Proportion odds model

(Lu and Zhang, 2007). RIDGE REGRESSION type method also has been extended to logistic regression (Le Cessie and Houwelingen, 1992; Zhu and Hastie, 2004). Different penalty (Fan and Li, 2001; Tibshirani, Saunders, Rosset, Zhu and Knight, 2005), negative log-likelihood of ℓ (Fan and Li, 2001; Park and Hastie 2007, Wang, Li and Tsai 2007, Wang and Leng, 2007) as well as better algorithms (Rosset and Zhu, 2007; Park and Hastie, 2007) also have been studied.

In this paper, we consider a general loss of differentiable ℓ, and a particular form of $J_\lambda(\beta) = \lambda\beta^T\Lambda W\beta$, where λ is a tuning parameter, Λ is a constraint matrix and W is a weighted matrix of a user's choice. Note that the differentiable ℓ is very general and has a wide scope of coverage including the aforementioned squared error loss and negative log-likelihood. Shortly we will see that the particular form of $J_\lambda(\beta)$ is not limited either. In Section 3.2 we will show that the optimization problem with this form has an equivalent solution for using the aforementioned (adaptive) L^1 penalty such as in (adaptive) LASSO, (adaptive) L^2 penalty such as (adaptive) RIDGE REGRESSION, and (adaptive) combined L^1 and L^2 penalty such as (adaptive) ELASTIC NET. Thus it is a very flexible and data-oriented method for using either one of the three penalties. In Section 3.3, we give an unified solution for this problem with methods for estimating the covariance of the estimate and selection of the tuning parameters. In Section 3.4, we propose a unified algorithm for the negative log-likelihood of ℓ when $n > p$ and $n << p$ or correlated data, and a sequential minimal optimization (SMO) algorithm to avoid the computations of an inverse of a huge matrix. In Section 3.5, we present a small simulation study to demonstrate the usefulness of our method, and in Section 3.6 we use datasets to illustrate the efficacy of our method. Finally, we present a short discussion in Section 3.7. We delay the proofs in the appendix.

3.2 The equivalent penalty for L^1, L^2 and their combination

Our proposed method is the following adaptive L^2-regularization problem: Suppose $\lambda \geq 0$, $\Lambda = \text{diag}\{\lambda_i\}$, $W = \text{diag}\{w_i\}$, and $\lambda_i, w_i > 0$ with $\sum_{i=1}^p \frac{1}{\lambda_i} w_i = p$,

$$\hat{\beta}_\lambda = \arg\min_{\beta,\Lambda} \ell(y, \beta) + \lambda \beta^T \Lambda W \beta. \tag{3.2}$$

In order to establish the equivalence, we first note that the following set up:

$$\ell(y, \beta) + \lambda_0 \beta^T W \beta + \lambda \beta^T \Lambda W \beta. \tag{3.3}$$

can be written as (3.2) by re-parameterization. That is, let $c_* = \sum_{j=1}^p \frac{1}{\lambda_0 + \lambda\lambda_j} w_j$, $\lambda_* = p/c^*$ and $\lambda_{j*} = c_*(\lambda_0 + \lambda\lambda_j)/p$. Then $\sum_{j=1}^p \frac{1}{\lambda_{j*}} w_j = p$. With $\Lambda_* = \text{diag}\{\lambda_{j*}\}$ (3.3) becomes $\ell(y, \beta) + \lambda_* \beta^T \Lambda_* W \beta$, which is a special case of (3.2).

We now establish the main result below, whose proof is in the Appendix.

Proposition 2. *Suppose that ℓ is differentiable, then for any given λ, there exists a γ such that the solution, $\hat{\beta}_{\lambda,\lambda_0}$, in the minimization problem of (3.3) over β and Λ is the same as that of*

$$\hat{\beta}_{\gamma,\lambda_0} = \arg\min_{\beta} \ell(y, \beta) + \lambda_0 \beta^T W \beta + \gamma \sum_{j=1}^p w_j |\beta_j|, \tag{3.4}$$

and vice versa.

The formulation (3.4) in Proposition 2 is interesting because if ℓ is a least square loss, and $W = I$, then it reduces to elastic net (Zou and Hastie 2005). If ℓ is negative likelihood as in GLM and $W = I$, then it reduces to Park and Hastie (2007). Thus in either case, (3.4) is a generalization of elastic net type of penalty in that it incorporates adaptiveness (the weights W). Proposition 2 also shows that if $\lambda_0 = 0$, then (3.2) is equivalent to adaptive L^1 penalty. On the other hand, Proposition 2 and the fact that (3.3) is a special case of (3.2) illustrate that a general adaptive L^2 penalty as in (3.2) unifies an adaptive L^1 penalty, adaptive L^2 penalty as well as the adaptive elastic net penalty types. Thus this general adaptive L^2 penalty method can deal with $n >> p$ or correlated data with variable selection. Hence it is a data-driven approach.

In the next section, we will derive a unified solution of the above optimization problem using quadratic approximation to the loss of ℓ by assuming that ℓ is twice differentiable. This assumption is generally not restrictive. For instance in likelihood approach, see, Zhang and Lu (2007) and Lu and Zhang (2007). When the loss is squared error (exactly quadratic), Qiu, Yin and Wang (2007) developed an efficient algorithm, which covered adaptive- LASSO, RIDGE REGRESSION as well as ELASTIC NET.

3.3 A UNIFIED SOLUTION

Suppose that the true parameter is β_0. For given y_i, x_i, λ, Λ and W, let $\Lambda^* = 2\lambda\Lambda W$, the solution of (3.2) can be approximately derived by the solution of a quadratic formula below (in obvious notation to suppress y in ℓ).

$$\ell(\beta_0) + \bigtriangledown\ell(\beta_0)^T(\beta - \beta_0) + \frac{1}{2}(\beta - \beta_0)^T \bigtriangledown^2 \ell(\beta_0)(\beta - \beta_0) + \frac{1}{2}\beta^T\Lambda^*\beta, \tag{3.5}$$

where $\bigtriangledown\ell(\beta_0) = \frac{\partial\ell(\beta_0)}{\partial\beta}$, $\bigtriangledown^2\ell(\beta_0) = \frac{\partial\ell^2(\beta_0)}{\partial\beta\partial\beta^T}$, and $\Lambda^* = 2\lambda\Lambda W$.

Then, the minimization of (3.5) is

$$\beta_1 = \beta_0 - (\bigtriangledown^2\ell(\beta_0) + \Lambda^*)^{-1}(\bigtriangledown\ell(\beta_0) + \Lambda^*\beta_0) = (\bigtriangledown^2\ell(\beta_0) + \Lambda^*)^{-1}(\bigtriangledown^2\ell(\beta_0)\beta_0 - \bigtriangledown\ell(\beta_0)). \tag{3.6}$$

An iteration algorithm such as Newton-Raphson method can be used here to find the final solution $\hat{\beta}$. Using techniques similar to those of Fan and Li (2001), we can approximate the covariance matrix of $\hat{\beta}$ by the following sandwich formula:

$$\widehat{Cov}(\hat{\beta}) = (\bigtriangledown^2\ell(\hat{\beta}) + \hat{\Lambda}^*)^{-1}\widehat{Cov}(\bigtriangledown\ell(\hat{\beta}))(\bigtriangledown^2\ell(\hat{\beta}) + \hat{\Lambda}^*)^{-1}, \tag{3.7}$$

where $\hat{\Lambda}^*$ is the estimate of Λ^* at $\beta = \hat{\beta}$.

In the next section, we will develop two algorithms and suggest the choices of λ, Λ and W.

3.4 Two Algorithms

In this section, we will propose two algorithms for our method: a unified algorithm (UA) and a general SMO algorithm (GSMOA). The unified algorithm shall work as long as we can

calculate the ℓ, such as negative log-likelihood for Cox' model (Tibshirani, 1997), proportional odds and hazard models (Lu and Zhang, 2007; Zhang and Lu, 2007), or generalized linear model (Park and Hastie, 2007). The general SMO algorithm is developed for avoiding the inversion of a huge matrix. We illustrate the UA approach with generalized linear model, and GSMOA approach for a logistic regression. Conditioning on x_i, y_i, a density is $f_i(y_i|g(x_i^T\beta))$, where g is a known link function. Thus the loss function, which is the conditional negative log-likelihood of y_i, is $\ell_i = -\log f_i$, and $\ell(\beta) = \sum_{i=1}^n \ell_i(g(X_i^T\beta), y_i)$.

3.4.1 Unified algorithm

Adaptive weights: Set $W = \text{diag}\{w_j\}$ where $w_j = |\hat{\beta}_j|^{-\nu}$ for some $\nu \geq 0$ and $\hat{\beta}_j$ is the GLM estimate. The choice of ν can follow from Zou (2006), and we use $\nu = 0, 0.5, 1$ in this paper for the purpose of comparison. Note that if $\nu = 0$, then $W = I$ throughout the algorithm, thus we refer it as the UA with equal weights.

Initial choices of the parameters: The initial tuning parameter can be selected by using MGCV (Wood, 2000) method. At convergence for given W, the MGCV-type statistic is constructed as

$$\text{MGCV}(\tilde{\Lambda}) = \frac{\ell(\hat{\beta})}{n[1 - p(\tilde{\Lambda})/n]^2}, \tag{3.8}$$

where $p(\tilde{\Lambda}) = \text{trace}[(\bigtriangledown^2\ell(\hat{\beta}) + \tilde{\Lambda}W)^{-1} \bigtriangledown^2 \ell(\hat{\beta})]$, and $\tilde{\Lambda} = 2\lambda\Lambda$.

If $n > p$, one can run GLM to obtain initial estimate of β; otherwise, when $n < p$, we enlarge the data by adding an identity $p \times p$ matrix for x and $p \times 1$ of zeros to y, so that the artificial data has sample size bigger than p, and we then run GLMPATH in R with $\lambda_1 = 0$ to obtain the initial estimate of β. This is allowed because we only use it for obtaining the initials of $\tilde{\Lambda}$, W and β which will be updated in the algorithm. Use the data (modified as described above if necessary), and set $W = I$, then obtain $\hat{\beta}$ by GLM and use MGCV to get $\hat{\tilde{\Lambda}}$. Thus set initial weights as $w_j^{(0)} = |\hat{\beta}_j|^{-\nu}$. Solve for $\lambda = \lambda^{(0)}$ and $\Lambda^{(0)}$ from $2\lambda^{(0)}\Lambda^{(0)} = \hat{\tilde{\Lambda}}\ \text{diag}\{\hat{\beta}_j^{-2}\}$, where $\Lambda^{(0)} = \text{diag}\{\lambda_j^{(0)}\}$ and $\sum_{j=1}^p \frac{1}{\lambda_j^{(0)}} w_j^{(0)} = p$. Based on Appendix A2, we propose the following algorithm for fixed $\lambda = \lambda^{(0)}$, accuracy δ and convergence criterion $\delta_0 = 10^{-6}$.

Unified Algorithm (UA).

1. Start $s = 0$, $p_0 = p$, $\Lambda^{(s)} = \Lambda^{(0)}$, $W^{(s)} = W^{(0)}$ and $\beta^{(-1)} = \hat{\beta}$. Set $\Lambda^* = 2\lambda\Lambda^{(s)}W^{(s)}$ and $\beta_0 = \beta^{(s-1)}$, find solution $\beta^{(s)}$ by equation (3.6) for p_s predictors.

2. For $j = 1, \cdots, p_s$, calculate $b_j^{(s)} = \frac{|\beta_j^{(s)}|}{\sum w_j^{(s)}|\beta_j^{(s)}|}$. If $b_j^{(s)} < \delta$, delete jth variable; if $b_j^{(s)} \geq \delta$, keep it.

3. Let p_{s+1} be the number of predictors the remain in the model. For those variables, update $W^{(s+1)} = \text{diag}\{w_j^{(s+1)}\}$ for $j = 1, ..., p_{s+1}$, by $w_j^{(s+1)} = 1/|\hat{\beta}_j^{(s+1)}|^\nu$ for $\nu \geq 0$. For $n > p_{s+1}$, $\hat{\beta}_j^{(s+1)}$ is the GLM estimate, while for $n < p_{s+1}$, $\hat{\beta}_j^{(s+1)}$ is updated by GLM-PATH in R with $\lambda_1 = 0$ with reduced data. Update $\Lambda^{(s+1)}$ by $\lambda_j^{(s+1)} = (\frac{p|\beta_j^{(s)}|}{\sum w_j^{(s+1)}|\beta_j^{(s)}|})^{-1}$.

4. Update $\beta^{(s+1)}$ by equation (3.6) for the reduced variables, based on $\beta_0 = \beta^{(s)}$ and $\Lambda^* = 2\lambda\Lambda^{(s+1)}W^{(s+1)}$.

5. Calculate $\max \frac{|\beta_j^{(s)} - \beta_j^{(s+1)}|}{1+|\beta_j^{(s+1)}|}$. If it is less than δ_0, stop; otherwise, go to Step 2 until it is convergent.

In the procedure of UA, once variable is deleted, it never enters again. This way, the computation becomes much more efficient. This unified method is robust and can be applied to either correlated predictors, $n > p$ or $n < p$ case.

For fixed tuning parameter λ, the results will be affected by the accuracy δ. It is important to select δ as well as the tuning parameter λ. We set $0 < \delta \leq 1$. The tuning parameter $\lambda \in (0, \Delta]$, where Δ is an integer ($\Delta = 200$ in our simulations). We select a series of λ and a series of δ in the respective intervals. Our algorithm finds the solution for each pair of (δ, λ) in a matrix. And finally, we select the best parameter pairs of (δ, λ) that has the minimum of BIC-type selection criterion (Schwarz, 1979):

$$\text{BIC}_\delta(\lambda) \quad = \quad 2\ell(\beta_\delta(\lambda) + \log(n)p_\delta(\lambda),$$

where $p_\delta(\lambda)$ is the number of nonzero coefficients in $\beta_\delta(\lambda)$, a simple estimate for the degrees of freedom (Zou, Hastie and Tibshirani 2007).

Figures 3.1 and 3.2 show typical BIC plots for model 1 with cases $n > p$ and $n < p$, respectively; while Figures 3.3 and 3.4 show the typical BIC plots for model 2 with cases $n > p$ and $n < p$, respectively. Note that the vertical line is the $\lambda = \lambda^{(0)}$ that is selected by MGCV method in the previous section, which seems always close to the best λ. Thus we use $\lambda = \lambda^{(0)}$ as the selected tunning parameter for λ and it works well in our limited simulations.

Our choice of initials for Λ seems work well and stable. Nevertheless we tried different initials as constant $\lambda_j^{(0)} = \sum w_j/p$, and $\lambda_j^{(0)} = 1/w_j^{(0)}$. They do not affect the final results in our simulations and data analysis.

Note that without loss of generality and to be consistent with the literature in the area, we can standardize the variable of x_i's for $i = 1, ..., p$ to have mean 0 and variance 1. In the previous sections, we don't have an 'intercept' parameter (β_0) in linear combination of X. However, with the intercept, one can easily adopt the same proofs and solutions by having a constant variable of $x_0 = 1$ with fixed weight of 0. Our algorithm works with intercept.

3.4.2 A general SMO algorithm

Nevertheless, when p is in thousands such as in microarray data analysis, an inverse of a huge matrix with dimension $p \times p$ need to be calculated. This computation is prohibitive. To avoid handling such cases, we adopted a Sequential Minimal Optimization (SMO) method used by Zhu and Hastie (2004) for logistic regression. Let X be a $n \times p$ data matrix for the predictors, and a be a $n \times 1$ vector. Based on the general SMO derivation in Appendix A3, we have

$$\beta = \Lambda^{*(-1)} X^T a. \tag{3.9}$$

For illustrative purpose, we use logistic regression, although our general SMO algorithm can be easily adapted to other generalized liner models. In such a case,

$$\ell_a = -\sum_{i=1}^{n} [(y_i + a_i) \log(y_i + a_i) + (1 - y_i - a_i) \log(1 - y_i - a_i)].$$

Thus we also need constraint to have appropriate ℓ_a,

$$0 < y_i + a_i < 1 \tag{3.10}$$

Hence, the general SMO algorithm is modified accordingly as follows.

General SMO Algorithm (GSMOA).

1. Choose a^0 satisfying conditions (3.18) and (3.10). Set $W^{(0)} = \Lambda^{(0)} = I$, $\beta^{(-1)} = I$, $r = s = 0$, and $p_0 = p$, but fix λ.

2. (a) If a^r satisfies (3.19), obtain $\beta^{(s)}$ by (3.9), denote $a^s = a^r$, and go to step 3.

   (b) If not, update $a^{r+1} = \tilde{a}(t)$ with $a^r = a$ according to (7.8)-(7.10), where t minimizes the dual $f(a)$ in the Appendix A3. Go back to step (a).

3. For $j = 1, \cdots, p_s$, calculate $b_j^{(s)} = \frac{|\beta_j^{(s)}|}{\sum w_j^{(s)} |\beta_j^{(s)}|}$. If $b_j^{(s)} < \delta$, delete jth variable; if $b_j^{(s)} \geq \delta$, keep it. $W^{(s)} = \text{diag}\{w_j^{(s-1)}\}$ and $w_j^{(s)} = |\beta_j^{(s-1)}|^\nu$ for $\nu \geq 0$.

4. Let p_{s+1} be the remain variables in the model. Update $W^{(s+1)}$ and $\Lambda^{(s+1)}$, where $\lambda_j^{(s+1)} = (\frac{|\beta_j^{(s)}|}{\sum w_j^{(s+1)} |\beta_j^{(s)}|})^{-1}$.

5. With updated X, $\Lambda^{*(s+1)} = 2\lambda\Lambda^{(s+1)}W^{(s+1)}$, go to step 2 with $a^r = a^s$ to obtain $\beta^{(s+1)}$.

6. Calculate max $\frac{|\beta_j^{(s)} - \beta_j^{(s+1)}|}{1+|\beta_j^{(s+1)}|}$. If it is less than δ_0, stop; otherwise, go to step 3 and update.

The tuning parameters are selected by the same BIC-type criterion in UA. Note that the GSMOA algorithm can not be used to estimate the intercept, β_0. Thus once we have the estimate for β with reduced data, we can call UA to have the respective estimates of β_0, β as well as the covariance matrix.

3.5 Simulations

In this section, we consider two representative examples for comparing different penalty methods: lasso type, adaptive lasso type (LSA; Wang and Leng, 2007) and the combination of lasso and ridge type (PH; Park and Hastie, 2007). The two examples are logistic

regression and Poisson log-linear example. To measure the prediction error of any fitted model, we compute the model error (ME)

$$\mathrm{ME}(\hat{\beta}) = E[g(\hat{\beta}^T X) - g(\beta^T X)]^2,$$

where g is the respective mean function for the models. For instance, if it is the logistic model, we have $\mathrm{ME}(\hat{\beta}) = E[\exp(\hat{\beta}^T X)/\{1+\exp(\hat{\beta}^T X)\} - \exp(\beta^T X)/\{1+\exp(\beta^T X)\}]^2$. The ME is estimated via 1000 Monte Carlo simulations. The median relative model error (MRME) is reported by the median of the ratio of the model error of the fitted model against that of MLE for $n > p$, and that of ridge type (with L^2 penalty only) for $n < p$, respectively. We also report the number of correct zero coefficients which present truly unimportant variables, and the number of incorrect zeros which present important variables erroneously left-out by the procedure. In addition, the median absolute deviation divided by .6745, denoted by SD, of the respective estimated coefficients can be regarded as the true standard error. The median of the respective estimated standard errors by formula (3.7), denoted by SD_m, and the median absolute deviation error of the respective standard errors divided by .6745, denoted by SD_{mad}, measure the overall performance of the standard error from the sandwich formula.

Example 1. The logistic regression model is used by Fan and Li (2001).

$$Y \sim Bernoulli(g(x^T\beta)), \tag{3.11}$$

where g is a link function and $g(u) = exp(u)/(1+exp(u))$. We consider two cases: $n > p$ and $n < p$.

For $n > p$, set $n = 200$ and $p = 8$. Let $\beta = (3, 1.5, 0, 0, 2, 0, 0, 0)^T$. The first six components of x follow from $N_6(0, \Sigma_x)$ where each element of Σ_x is $\rho^{|i-j|}$ for all i and j, with $\rho = 0.5$. The last two components of x follow from Bernoulli distribution with probability of success 0.5 and they are independent to each other.

For $n < p$, set $n = 50$. Let $p = 100$ by adding 92 zeros to the β in $n > p$ case. The first eight x variables are the same as in case of $n > p$. The additional 92 components of x follow Bernoulli distribution with probability of success 0.5.

Table 3.1 reports the performance of UA approach with that of LASSO, LSA and PH in terms of MRME, the average number of correct zeros and the average number of incorrect zeros. For $n > p$, all UA (with different weights) have smaller MRME and bigger number of correct zero than all other three methods. Although UA has some positive incorrect zeros, it seems not significant. When $n < p$, PH is the only other method that is applicable. Again we conclude the same conclusion. It seems that UA with equal weights is the best method. To test the accuracy of formula (3.7), Table 3.2 reports the respective results. For $n > p$, all UA are rather accurate. For $n < p$, the results get a little bit worse, nevertheless the comparison between the true and estimate are within one stand deviation. Again UA with equal weights seems the best method.

Example 2. The Poisson log linear model provided by Fan and Li (2001) is

$$Y \sim Poisson(\lambda(x^T\beta)), \tag{3.12}$$

where λ is a link function and $\lambda(u) = exp(u)$. X is same as in example 1 except $\beta = (1.2, 0.6, 0, 0, 0.8, 0, 0, 0)^T$. The n and p are the same as in example 1 for both cases respectively.

Table 3.3 indicates that in both $n > p$ and $n < p$, all UAs with different weights beat other applicable methods while UA having equal weights is the best method. Table 3.4 reports the accuracy of the sandwich formula, again UA with equal weights is the best method, while others do have reasonable estimates. Overall UA with equal weights is the best method confirming our findings in example 1.

3.6 Two datasets

In this section, we illustrate our method via breast cancer data and microarray data.

A breast cancer data. A breast cancer biopsy data downloaded from R package MASS was used to test the methods among LAS, PH and UA. This data set was originally obtained from he University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. The

predictors are clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei (16 values are missing), bland chromatin, normal nucleoli, and Mitoses. The response variable is class of tumor which has two levels "benign" or "malignant". There are 699 patients involved in this study. However, 16 values are missing for variable bare nuclei, these observations are not involved in the later analysis. Thus, we only use 683 observations in the analysis below.

The breast cancer data were randomly divided into two roughly equal parts. The first part were used to do model fitting and tuning parameter selection by tenfold CV or MGCV and carry out LSA, PH and UA. The second part is validation part. Errors-in-sample was calculated with first part of data and errors-out-sample was calculated with second part of data. We repeated the procedure for 100 times.

Table 3.5 shows that in terms of errors, UA with equal weights and UA with weight of $\nu = .5$ are the best method, UA with weight of $\nu = 1$ is the second best, and all three outperform PH and LSA. The results confirm the finding in our simulations. Table 3.6 shows the variable selections, LSA seems choose the important variables very differently from PH and UAs. While PH and UAs are pretty much agree with each other. Particularly for PH and UA with equal weights.

Combing the information from both tables, we think UA with equal weights provides an important procedure.

Leukaemia cancer data. Leukaemia cancer gene expression data by Golub et al.(1999) has been analyzed by Park and Hastie (2007) with L_1-regulation path algorithm. There are 72 observations on 7129 gene expression measurements and an indicator variable of type of genes. The data has been divided into validating set (38) and testing set (34). To make a comparison between UA with SMO algorithm and previous methods (Golub et al. 1999); Tibshirani et al. (2005); Zhu and Hastie (2004); Park and Hastie (2007), cross-validation error, test error and number of gene selected are calculated. Table 3.7 indicates that UA yielded the best cv error and the best test error along with PH and Zhu and Hastie (2004).

The number of selected genes are 24, only one more than PH but 2 less than Zhu and Hastie (2004). Overall again this dataset demonstrates that UA is a useful method.

3.7 Discussion

In this paper, we extended generalized adaptive ridge regression to a general loss set up. We proved the equivalence between adaptive L_1 and constrained adaptive L_2 penalties in the general loss minimization problems, and developed a unified algorithm as well as a generalized SMO algorithm for avoiding the inversion of a huge matrix. The basic equivalence also can be extended to other setting such as Wang and Leng (2007), which may improve their algorithm as well. The simulations as well as data analysis seem illustrate that our approach provides an important alternative for some available methods.

Appendix: Justification

A1. Proof of Proposition 2. Define for $j = 1, \cdots, p$, $c_j = \sqrt{\lambda_j}\beta_j; b_j = \sqrt{\frac{1}{\lambda_j}}$, we have $\sum_{j=1}^p w_j b_j^2 = p$, and $c_j b_j = \beta_j$. Write $\ell(\beta) = \sum_{i=1}^n \ell_i(\beta)$, Equation (3.3) then becomes

$$\sum_{i=1}^n \ell_i(c_1 b_1, ..., c_j b_j, ..., c_p b_p) + \lambda_0 \sum_{j=1}^p w_j c_j^2 b_j^2 + \lambda \sum_{j=1}^p w_j c_j^2,$$

under $\sum_{j=1}^p w_j b_j^2 = p$. To solve this minimization problem, we have

$$\sum_{i=1}^n \ell_i(c_1 b_1, ..., c_j b_j, ..., c_p b_p) + \lambda_0 \sum_{j=1}^p w_j c_j^2 b_j^2 + \lambda \sum_{j=1}^p w_j c_j^2 + m(\sum_{j=1}^p w_j b_j^2 - p).$$

Let $S_k = \frac{\partial \sum_{i=1}^n \ell_i(c_1 b_1, ..., c_j b_j, ..., c_p b_p)}{\partial (c_k b_k)}$, by taking derivatives with respect to c_k and b_k, we have the following equations:

$$c_k S_k + 2\lambda_0 w_k c_k^2 b_k = -2m w_k b_k,\ b_k S_k + 2\lambda_0 w_k c_k b_k^2 = -2\lambda w_k c_k.$$

We have either $c_k = 0$ which leads to $\beta_k = 0$; or $c_k \neq 0$. In the latter case, the two equations become

$$S_k + 2\lambda_0 w_k c_k b_k = -2m w_k b_k / c_k,\ S_k + 2\lambda_0 w_k c_k b_k = -2\lambda w_k c_k / b_k.$$

The product of these two equations leads to $\frac{(S_k+2\lambda_0 w_k c_k b_k)^2}{w_k^2} = 4\lambda m = S_0^2$, where $S_0 = (S_k + 2\lambda_0 w_k c_k b_k)/w_k$. Hence, the first equation leads to $c_k S_0 = -2mb_k$. Multiple both sides by $w_k b_k$, we have $w_k b_k c_k S_0 = -2m w_k b_k^2$ which leads to $w_k \beta_k S_0 = -2m w_k b_k^2$. Thus $w_k|\beta_k||S_0| = 2m w_k b_k^2$, by the fact that $\sum_{j=1}^p w_j b_j^2 = p$, we have $|S_0| \sum_{j=1}^p w_j|\beta_j| = 2mp$. And similarly for the other equation, we have $|S_0|p = 2\lambda \sum_{j=1}^p w_j|\beta_j|$. Thus

$$m = \frac{\lambda(\sum_{j=1}^p w_j|\beta_j|)^2}{p^2}. \tag{3.13}$$

The fact that the left hand sides of the two equations are the same, and by the definitions of c_k and b_k, we have

$$\frac{b_k}{c_k} = \sqrt{\frac{\lambda}{m}} sign(\beta_k).$$

Put it into $S_k+2\lambda_0 w_k c_k b_k = -2m w_k \frac{b_k}{c_k}$, we have $S_k+2\lambda_0 w_k \beta_k+2m w_k \sqrt{\frac{\lambda}{m}} sign(\beta_k) = 0$. Using (3.13) we have $S_k + 2\lambda_0 w_k \beta_k + 2w_k \frac{\lambda}{p} sign(\beta_k) \sum_{j=1}^p w_j|\beta_j| = 0$. That is, the final solutions satisfy either $\beta_k = 0$, or

$$S_k + 2\lambda_0 w_k \beta_k + 2w_k \frac{\lambda}{p} sign(\beta_k) \sum_{j=1}^p w_j|\beta_j| = 0, \tag{3.14}$$

which however, is the solution of the following minimization problem:

$$\hat{\beta} = \arg\min \sum_{i=1}^n \ell_i(\beta) + \lambda_0 \beta^T W \beta + \frac{\lambda}{p}(\sum_{j=1}^p w_j|\beta_j|)^2,$$

which is equivalent to $\hat{\beta} = \arg\min \sum_{i=1}^n \ell_i(\beta) + \lambda_0 \beta^T W \beta$ under $\frac{\lambda}{p}(\sum_{j=1}^p w_j|\beta_j|)^2 < t'$, which is then equivalent to $\hat{\beta} = \arg\min \sum_{i=1}^n \ell_i(\beta) + \lambda_0 \beta^T W \beta$ under $\sum_{j=1}^p w_j|\beta_j| < t$ for some positive t. The last minimization problem is equivalent to (3.4) for some γ. Hence we proved that the two solutions are the same.

A2. Algorithm background. Using the notation in the proof of Proposition 2, from the proof, $b_k^2 = c_k^2 \frac{\lambda}{m} = \frac{c_k^2 p^2}{(\sum w_j|\beta_j|)^2}$, implies that $b_k^4 = \frac{b_k^2 c_k^2 p^2}{(\sum w_j|\beta_j|)^2} = \frac{\beta_k^2 p^2}{(\sum w_j|\beta_j|)^2}$. Hence $b_k^2 = \frac{p|\beta_k|}{\sum w_j|\beta_j|}$. Thus

$$\lambda_k = \frac{1}{b_k^2} = (\frac{p|\beta_k|}{\sum w_j|\beta_j|})^{-1}, \tag{3.15}$$

where the constraint on Λ is automatically satisfied. Therefore, an equivalent algorithm can be developed iteratively between (3.6) and (3.15). That is, giving λ, W, and initial β_0, if λ_k's are given, find β by (3.6); check b_k^2 or $b_k^2/p = \frac{|\beta_k|}{\sum w_j|\beta_j|}$ if it is smaller than a threshold, delete the respective β_k; otherwise keep it there. Then for the remain variables, update the respective λ_k by (3.15). Proceed this scheme iteratively until it is convergent. □.

A3. Derivation of SMO formula. Note that our general minimization can be written as

$$\sum_{i=1}^{n} \ell(\xi_i) + \frac{1}{2}\beta^T \Lambda^* \beta, \text{ under } \xi_i = \beta_0 + X_i^T \beta.$$

The Lagrangian of this problem is

$$L = \sum_{i=1}^{n} \ell(\xi_i) + \frac{1}{2}\beta^T \Lambda^* \beta + \sum_{i=1}^{n} a_i(\xi_i - \beta_0 - X_i^T \beta).$$

Thus the optimization equations with $X^T = (X_1^T, ..., X_n^T)$ and $a^T = (a_1, ..., a_n)$ are $\ell'(\xi_i) + a_i = 0; \sum_{i=1}^n a_i = 0; \Lambda^* \beta = X^T a$. Thus,

$$\begin{aligned} \xi_i &= \ell'^{(-1)}(-a_i) && (3.16) \\ \beta &= \Lambda^{*(-1)} X^T a. && (3.17) \\ \sum_{i=1}^{n} a_i &= 0; && (3.18) \end{aligned}$$

And thus,

$$L = \sum_{i=1}^{n} \ell(\ell'^{(-1)}(-a_i)) - \frac{1}{2}\beta^T \Lambda^* \beta + \sum_{i=1}^{n} a_i \ell'^{(-1)}(-a_i).$$

Finally, apply the Wolfe duality theory, we have to minimize

$$f(a) = \ell_a + \frac{1}{2} a^T X \Lambda^{*(-1)} X^T a, \text{ under (3.18)}$$

where $\ell_a = -\sum_{i=1}^n [\ell(\ell'^{(-1)}(-a_i)) + a_i \ell'^{(-1)}(-a_i)]$, and $\ell'(.)$ is the respective derivative. Once a solution of a is obtained, β can be obtained via (3.17).

The Lagrangian for the new optimization problem becomes

$$\tilde{L} = f(a) - \tau \sum_{i=1}^{n} a_i.$$

Let $\mathbf{e}_i$ be the unit vector with ith element being 1; otherwise, 0. Define

$$F_i = \frac{\partial \ell_a(a_i)}{\partial a_i} + a^T X \Lambda^{*(-1)} X^T \mathbf{e}_i.$$

Then the optimality condition is $\frac{\partial \tilde{L}}{\partial a_i} = F_i - \tau = 0, i = 1, ..., n$; which is equivalent to

$$\max_i F_i = \min_i F_i. \tag{3.19}$$

Let $i_u = \arg\max F_i$ and $i_l = \arg\min F_i$. Then if (3.19) doesn't hold, then define

$$\tilde{a}_{i_u}(t) = a_{i_u} + t, \tag{3.20}$$

$$\tilde{a}_{i_l}(t) = a_{i_l} - t, \tag{3.21}$$

$$\tilde{a}_i(t) = a_i, \text{ if } i \neq i_u, i_l. \tag{3.22}$$

and

$$\frac{\partial f(a(t))}{\partial t} = F_{i_u} - F_{i_l}$$

where F_{i_u} and F_{i_l} are evaluated at $\tilde{a}(t)$, which is not 0 at $t = 0$. A decrease in f is possible by choosing t suitably away from 0.

3.8 References

[1] Cessie, S. and Houwelingen J.C. (1992) Ridge estimators in logistic regression.*Appl. Statist.* 41, 191-201.

[2] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96, 1348-1360.

[3] Golub et al. (1999) Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science* 286, 531538.

[4] Hoerl, A.E. and Kennard, R.W. (1970a). Ridge regression: applications to nonorthogonal problems. *Technometrics*, 12, 69-82.

[5] Hoerl, A.E. and Kennard, R.W. (1970b). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12, 55-67.

[6] Lu, W. and Zhang, H. (2007). Variable selection for proportional odds model. *Statistics in Medicine* 26, 3771-3781.

[7] Park, M. Y. and Hastie, T. (2007) An L1 regularization-path algorithm for generalized linear models. *J. R. Statist. Soc. B* 69, 659-677.

[8] Qiu, J. Yin, X. and Wang, H. (2008). Generalized Adaptive Ridge: a Data Driven Approach. Submitted.

[9] Rosset, S. and Zhu, J. (2007). Piecewise linear regularized solution paths. *Annals of Statistics*, 35, 1012-1030.

[10] Schwarz, G. (1979). Estimating the dimension of a model. *Annals of Statistics*, 6, 461-464.

[11] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B* 58, 267-288.

[12] Tibshirani, R. (1997). The lasso method for variable selection in the Cox model. *Statistics in Medicine* 16, 385395.

[13] Tibshirani, R. Saunders, M. Rosset, S. Zhu, J. and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *J. R. Statist. Soc. B* 67, 91108.

[14] Wang, H. and Leng, C. (2007). Unified LASSO estimation by least Squares Approximation. *Journal of the American Statistical Association*, 102, 1039-1048.

[15] Wang, H., Li, G. and Tsai, C. L. (2007). Regression coefficient and autoregressive order shrinkage and selection via lasso. *Journal of Royal Statistical Society, Series B*, 69, 63-78.

[16] Wood, S.N. (2000). Modelling and Smoothing Parameter Estimation with Multiple Quadratic Penalties. *J.R.Statist.Soc.B* 62(2), 413-428

[17] Zhang, H. and Lu, W. (2007). Adaptive Lasso for COX's Proportional Hazards Model. Biometrika 1-17.

[18] Zhu, J. and Hastie, T. (2004). Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 5, 427-443.

[19] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 67, 301-320.

[20] Zou, H. (2006). The adaptive-LASSO and its oracle properties. *Journal of American Statistical Association* 101, 1418-1429.

[21] Zou, H., Hastie, T. and Tibshirani, R. (2007). On the 'degrees of freedom' of lasso. *The Annals of Statistics*, 35, 2173-2192.

Table 3.1: Simulation results for logistic regression of Example 1

Method	MRME(%)	Avg. No. of 0 coeff.		MRME(%)	Avg. No. of 0 coeff.	
		correct	incorrect		correct	incorrect
LASSO	53.14	3.76	0			
LSA	50.4	4.82	0			
PH	54.4	4.73	0	42.13	95.55	0.0
UA($\nu = 0$)	26.32	4.99	0.02	37.63	96.75	0.05
UA($\nu = 0.5$)	33.07	4.85	0	36.09	96.15	0.4
UA($\nu = 1$)	33.18	4.88	0.01	34.03	96.4	0.3

The left part in the table is for $n > p$, the right part in the table is for $n < p$.

Table 3.2: Standard deviations of estimators for logistic regression of Example 1

	$\hat{\beta}_1$		$\hat{\beta}_2$		$\hat{\beta}_5$	
Method	SD	$SDm(SDmad)$	SD	$SDm(SDmad)$	SD	$SDm(SDmad)$
LASSO	0.310	0.379(0.037)	0.285	0.284(0.019)	0.244	0.287(0.019)
UA($\nu = 0$)	0.449	0.443(0.064)	0.321	0.320(0.035)	0.308	0.343(0.055)
UA($\nu = 0.5$)	0.480	0.494(0.078)	0.381	0.357(0.050)	0.372	0.370(0.058)
UA($\nu = 1$)	0.543	0.503(0.097)	0.448	0.370(0.070)	0.373	0.385(0.064)
UA ($\nu = 0$)	0.962	0.911(0.313)	0.886	0.894(0.373)	0.820	0.740(0.249)
UA($\nu = 0.5$)	1.370	1.238(0.478)	0.932	0.640(0.464)	1.168	0.902(0.258)
UA($\nu = 1$)	1.854	1.418(0.924)	1.206	1.185(0.560)	1.267	1.092(0.298)

The upper part in the table is for $n > p$, the lower part in the table is for $n < p$.

Table 3.3: Simulation results for Poisson regression of Example 2

Method	MRME(%)	Avg. No. of 0 coeff.		MRME(%)	Avg. No. of 0 coeff.	
		correct	incorrect		correct	incorrect
LSA	50.2	4.79	0			
LASSO	59.92	3.68	0			
PH	55.6	4.84	0	31.06	94.75	0.58
UA($\nu = 0$)	19.64	5	0	11.14	96.95	0.5
UA($\nu = 0.5$)	29.80	5	0	12.57	96.90	0.5
UA($\nu = 1$)	31.05	5	0	32.96	96.80	0.85

The left part in the table is for $n > p$, the right part in the table is for $n < p$.

Table 3.4: Standard deviations of estimators for poisson regression of Example 2

	$\hat{\beta}_1$		$\hat{\beta}_2$		$\hat{\beta}_5$	
Method	*SD*	*SDm*(*SDmad*)	*SD*	*SDm*(*SDmad*)	*SD*	*SDm*(*SDmad*)
LASSO	0.086	0.078(0.013)	0.101	0.082(0.069)	0.083	0.074(0.017)
UA($\nu = 0$)	0.039	0.037(0.007)	0.038	0.036(0.007)	0.026	0.024 (0.007)
UA($\nu = 0.5$)	0.039	0.032(0.006)	0.040	0.044(0.009)	0.031	0.040(0.006)
UA($\nu = 1$)	0.037	0.035(0.007)	0.043	0.037(0.005)	0.040	0.031(0.007)
UA($\nu = 0$)	0.119	0.076(0.049)	0.093	0.082 (0.014)	0.120	0.108(0.018)
UA($\nu = 0.5$)	0.178	0.096(0.087)	0.163	0.109(0.069)	0.142	0.111(0.043)
UA($\nu = 1$)	0.201	0.150(0.098)	0.203	0.167(0.084)	0.189	0.136(0.062)

The upper part in the table is for $n > p$, the lower part in the table is for $n < p$.

Table 3.5: Cancer data: comparison among LSA, PH and UA

	Errors in-sample	Errors out-sample
LSA	20/342	23/341
PH	16/342	20/341
UA($\nu = 0$)	10/342	12/341
UA($\nu = 0.5$)	10/342	12/341
UA($\nu = 1$)	11/342	13/341

Table 3.6: Cancer data: percentage of variable selected over 100 runs by LSA, PH and UA

	Var1	Var2	Var3	Var4	Var5	Var6	Var7	Var8	Var9
LSA	0.4	0.1	0.3	0.3	0.05	0.6	0.35	0.1	0
PH	0.9	0.5	0.95	0.75	0.75	1	0.85	0.85	0.5
UA($\nu = 0$)	1	0.15	0.85	0.75	0.1	1	0.8	0.55	0.6
UA($\nu = 0.5$)	0.6	0.15	0.55	0.5	0.05	0.6	0.6	0.45	0.35
UA($\nu = 1$)	0.8	0.15	0.6	0.7	0.25	0.85	0.7	0.55	0.65

Table 3.7: Leukemia data: comparison between previous methods and UA

Methods	CV error	test error	selected genes
Golub et al.(1999)	3/38	4/34	50
Tibshirani et al. (2002)	2/38	2/34	21
Zhu et al. (2004)	2/38	1/34	26
Park et al. (2007)	2/38	1/34	23
UA($\nu = 0$)	2/38	1/34	24

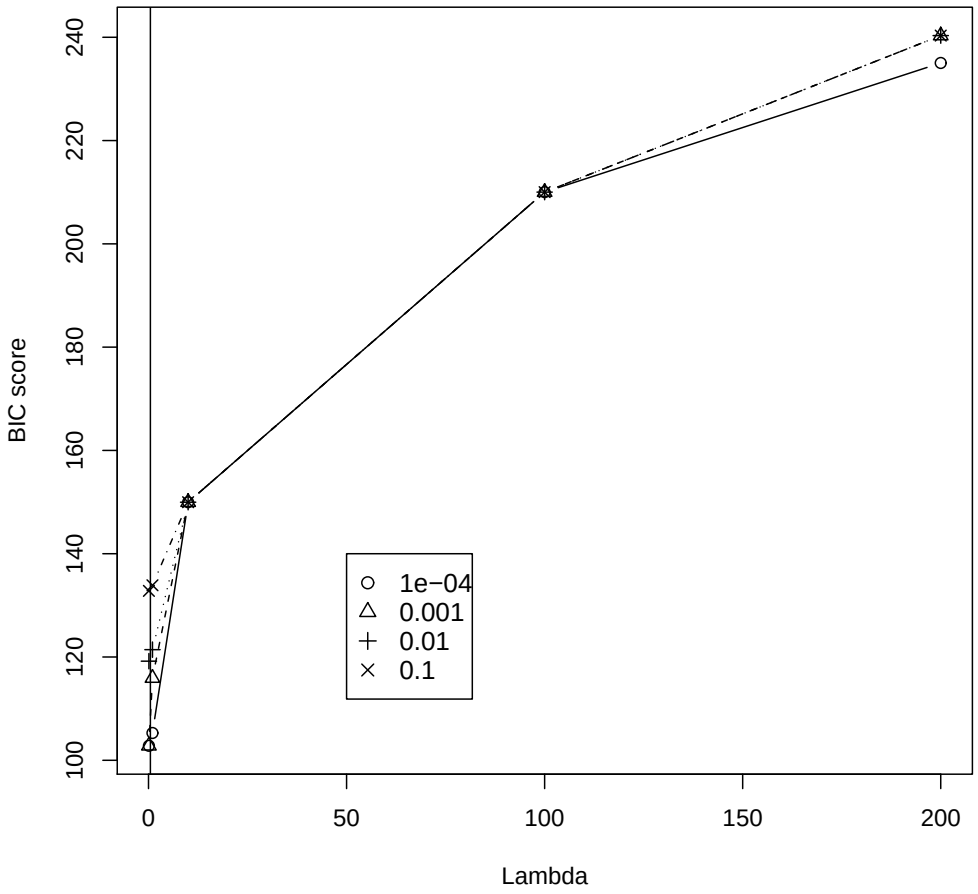

Figure 3.1: Tuning parameters selection (BIC) for Logistic Model ($n > p$)

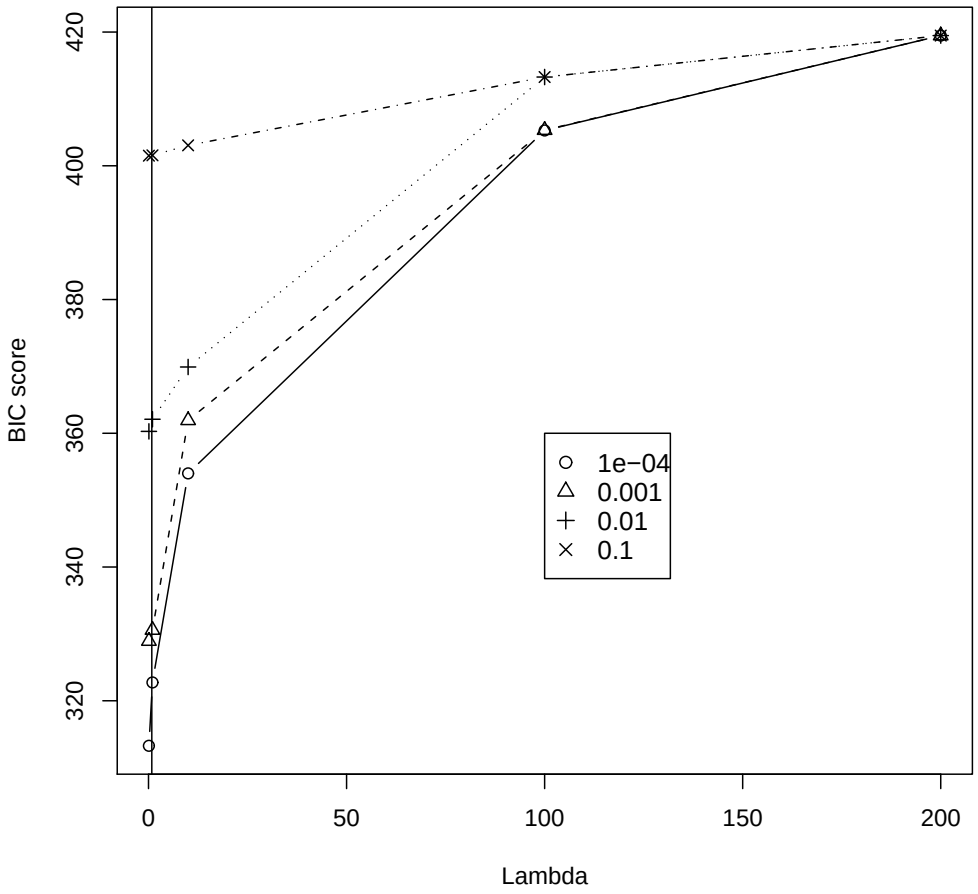

Figure 3.2: Tuning parameters selection (BIC) for Logistic Model ($n < p$)

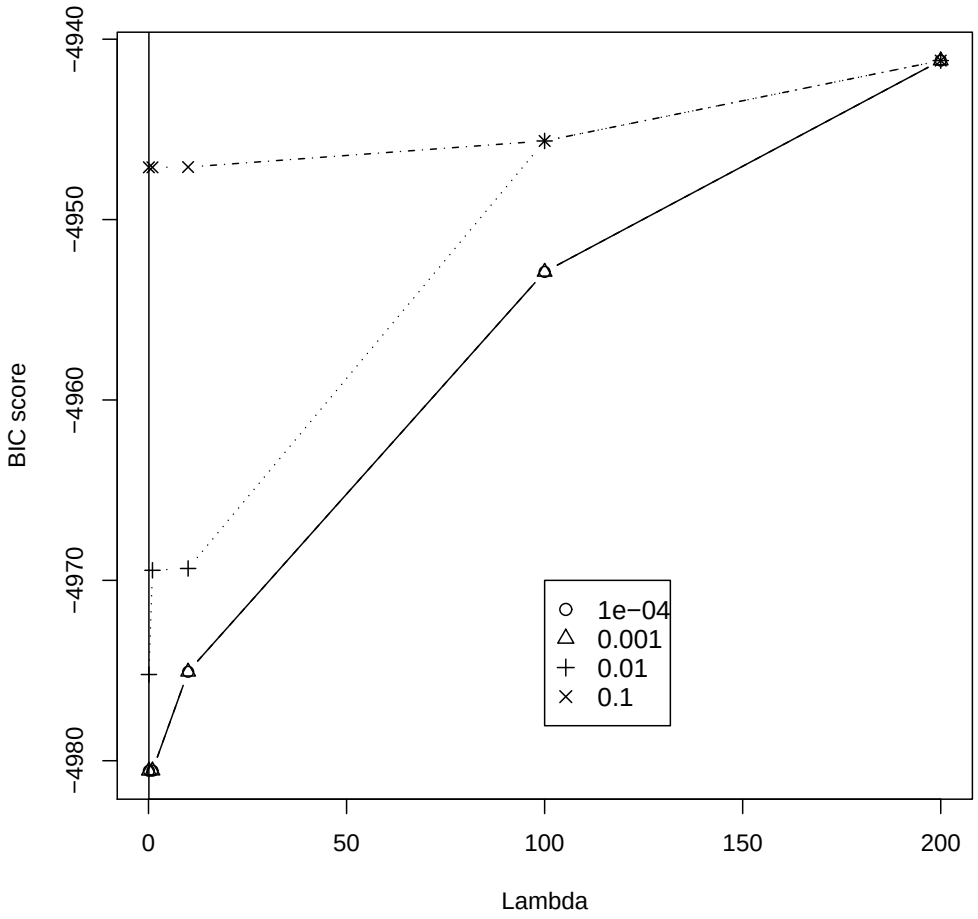

Figure 3.3: Tuning parameters selection (BIC) for Poisson Model ($n > p$)

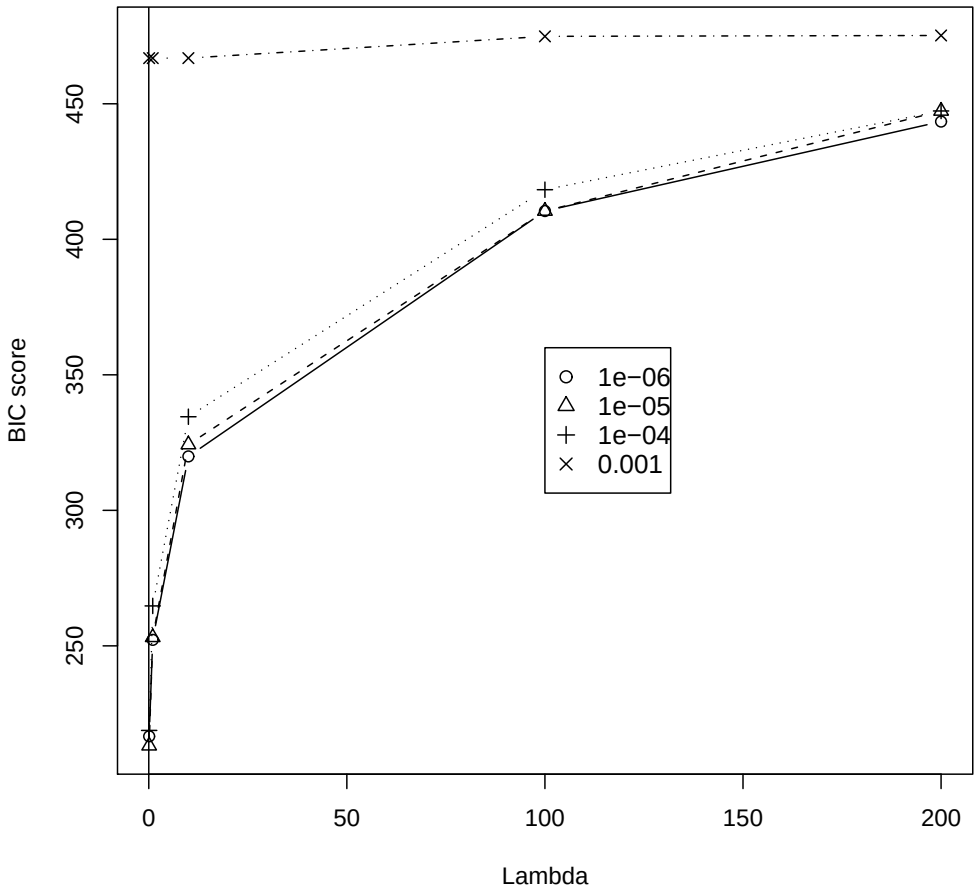

Figure 3.4: Tuning parameters selection (BIC) for Poisson Model ($n < p$)

Chapter 4

A variable selection approach to multi-dimensional NMR spectra interpretation[1]

[1]Qiu, J., Yin, X. and Tian, F. To be Submitted to *Journal of Magnetic Resonance.*

Abstract

Three-way decomposition (TWD) is a very versatile analysis tool with applications in a variety of protein nuclear magnetic resonance (NMR) fields. It has been used to extract structural data from 3D NOESYs, to determine relaxation rates in a large proteins, to identify ligand binding in screening for lead compounds, and to complement non-uniformly recorded (sparse) spectra. For instance, multi-dimensional NMR spectra interpretation (MUNIN) (Orekhov et al., 2001) is a method based on TWD, for the automated interpretation of three-dimensional NMR spectra. An NMR spectra is decomposed into sum of components, with each component corresponding to one or a group of peaks. Each component is defined as the direct product of three one dimensional shapes. A consequence is reduction in dimensionality of the spectral data used in further analysis. The decomposition may be applied to frequency-domain or time-domain data or mixture of them. MUNIN has good features, however, improvements still can be done via the study of accuracy and robustness by introducing a penalty term of quadratic form (Luan et al., 2005). Nevertheless, both methods don't have statistical approach to estimate the number of true terms in the decomposition. In the spirit of recently developed generalized adaptive ridge (GAR), we propose a new method by combining GAR and TWD to form an adaptive three-way decomposition (ATWD) for analyzing NMR data. The general penalty term makes the methods developed by Luan et al. (2005) and MUNIN as special cases of ATWD. In addition, ATWD inherits the good features of MUNIN as well as statistically estimate the number of true terms. Thus ATWD improves both previous methods. The applications of this method were illustrated in both simulation studies and real data analysis.

Key Words: Generalized adaptive ridge; MUNIN; NMR; Penalization; Relaxation; Three-way decomposition; Variable selection

4.1 Introduction

NMR plays an important role in chemistry and other sciences. The appearance of Fourier transform and multi-dimensional NMR makes this tool more valuable for studies on complex chemicals, especially biological macromolecules (Malmodin and Billeter, 2005). With the development of hardware, experimental techniques and advances in protein studies, the complexity of NMR data has been increasing (Stern et al., 2002). NMR spectroscopy is a concept which decomposes experimental input, raw time-domain data, into components that correspond to a single peak or a group of peaks. The estimation of features including frequencies and amplitudes is the key to separate peaks in an NMR spectrum. Basically, there are two steps to obtain a NMR spectroscopy (Orekhov et al., 2001). The first step is to apply discrete Fourier transform (DFT) to the raw time-domain NMR data. The second step is to identify the individual signals via interactive or automated peak picking procedures.

Even though DFT is a well-established method which is fast and robust, it has some drawbacks while being applied to multi-dimensional NMR spectroscopy. DFT requires experimental data input to be sampled at regular time intervals. That means to optimize data sampling in terms of sensitivity and resolution for a given duration of a NMR experiment is impossible. DFT is most powerful for peak separation if frequency differences are the major concern. However, if line shapes influenced mainly by relaxation decay or by J-coupling, DFT is powerless (Orekhov et al., 2001).

Linear prediction prior to Fourier transformation (Koehl, 1999) has been widely used to extrapolate and reconstruct the measured signal. Another modern but less widely used spectra analysis method is maximum-entropy reconstruction (Hoch and Stern, 1996; Schmieder et al., 1997). Comparisons between these methods were made in terms of accuracy and precision. In most cases, maximum-entropy method performs better than linear prediction, with respect to accuracy of spectra, sensitivity and resolution, and false-negative peaks. In addition, maximum-entropy method relies on few assumptions, handles data sampled irregularly and data with missing values. However, peak shapes could be distorted and

peak intensity could be changed by maximum-entropy method. All of the methods discussed above can process in practice only one- or two-dimensional NMR data sets at any given time.

TWD is a mathematical concept (Carroll and Chang, 1970; Harshamn, 1970) and it is a popular tool for analyzing various types of NMR data sets (Luan et al., 2005). TWD is also called parallel factor analysis (PARAFAC) or canonical decomposition (Carroll and Pruzansky, 1984). Before TWD is applied to NMR spectroscopy (Orekhov et al., 2001; Luan et al., 2005), it has been applied in fields like chemometrics or psychometrics (Harshman and Lundy, 1984). A method called Multi-dimensional NMR spectra interpretation (MUNIN) was developed by combining the concept of TWD and a simple model for NMR spectra (Orekhov et al., 2001). MUNIN can group signals found in NMR spectrum into subsets called components. MUNIN can resolve overlapped signals efficiently. This approach tries to avoid inclusion of noise and certain artifacts into the components to some extent and let them collected in residual terms. One of advantages for MUNIN is that it dose not require any assumptions on signal shapes. In that way, MUNIN can handle signals with different shapes equally well. In addition, different shape types along various dimensions of multidimensional spectra can be combined arbitrarily. MUNIN analysis results are very helpful for structure analysis. MUNIN method has also been applied to relaxation studies and reconstruction of time-domain data sets from sparse experimental data. MUNIN can be treated as a complement to traditional spectra data analysis method for high-dimensional spectra data. MUNIN is also very efficient for reducing the dimensions of multi-dimensional data sets.

Although MUNIN using TWD is very useful, statistically, it uses the least squares formula to estimate parameters. Hence, improvements using penalty terms can be applied. In fact, based on ridge regression idea, Luan et al. (2005) proposed a new approach which includes a quadratic penalty term into the objective function for minimization. The penalty term can ensure that amplitudes of all components are of comparable size. In addition, simulation

results indicated that TWD method with a quadratic penalty term tends to improve precisions and accuracies even under the conditions of weak signals, higher overlap and lower signal noise ratio. This method has power to handle overlapped signals and data with low signal noise ratio to some extend. Nevertheless, a typical ridge type penalty may improve accuracy but may not be able to help selecting the correct number of signals (the number of terms in the decomposition). Therefore, further improvements can be made by using variable selection ideas. In this paper, we use a general adaptive quadratic penalty idea that was studied by Qiu, Yin and Wang (2008) to improve accuracy as well as select the number of signals. Our method includes both MUNIN and the method proposed by Luan et al. (2005) as special cases. Therefore, it is more general. The exact form of the general adaptive quadratic penalty term is determined by the experimental data used in the analysis, making this novel method a data driven approach. We will use simulations to demonstrate the efficacy of our approach adopting the comparison scheme that was used by Luan et al. (2005), and one real data to illustrate its usefulness.

4.2 Method

In this section, we first review some similar existing methods, then we propose our approach.

4.2.1 One-dimensional NMR data analysis

Suppose that one-dimensional NMR data $y(t)$ is observed, and assume that the shape of signal $y(t)$ is exponentially decaying sinusoids. Therefore, each one-dimensional NMR data point can be decomposed into a sum of one dimensional signals as follows.

$$y(t) \quad = \quad \sum_{j=1}^{M} A_j F1_j + \epsilon_t, \tag{4.1}$$

where $y(t)$ is the signal observed at time t, M is the number of components, A is amplitude of the signal, F1 is the shape of signal and ϵ_t is the noise which is assumed to follow standard normal distribution. Furthermore, by assuming the shape of signal as exponentially decaying

sinusoids, $F1_j = \exp(\frac{-t}{T_j})(\cos(\omega_j t) + i\sin(\omega_j t))$, equation (4.1) can be rewritten as

$$y(t) = \sum_{j=1}^{M} A_j \exp(\frac{-t}{T_j})(\cos(\omega_j t) + i\sin(\omega_j t)) + \epsilon_t, \tag{4.2}$$

where T_j is the parameter related to decay rate and ω_j is frequency. Assume that the number of component M is known, the parameters in equation (4.1) can be estimated by least-square minimization (Orekhov et al., 2001). That is,

$$MIN_{(A,F1,M)} \sum_{t=1}^{R} (y(t) - \sum_{i=j}^{M} A_j F1_j)^2, \tag{4.3}$$

where R is the number of sampling.

Least squares minimization optimization also can be used for estimating the parameters in equation (4.2). That is,

$$MIN_{(A,T,\omega,M)} \sum_{t=1}^{R} (y(t) - \sum_{j=1}^{M} A_j \exp(\frac{-t}{T_j})(\cos(\omega_j t) + i\sin(\omega_j t)))^2. \tag{4.4}$$

This method is simple as long as M is known. However, practically one has to estimate M. Due to the noise in real data, the estimated number of components $\hat{M}$ with ordinary least square method may be bigger than the number of signals. Thus, in order to reduce the effects of noise, accurately estimate the parameters including correctly identify M, dimension reduction or variable selection methods can be used to reduce the inflation of number of signals.

4.2.2 Multi-dimensional NMR data analysis

One-dimensional NMR data analysis as we discussed in the previous section cannot handle complex systems such as proteins and DNA, because experimental data are usually multi-dimensional. However, the estimation idea can be adopted for multi-dimensional NMR data analysis.

Suppose there is a three-dimensional NMR data $y(i, j, k)$. Without any assumption on the shapes of signals at each dimension, the observed NMR spectra is decomposed into the sum of all signals plus some measurement error. That is,

$$Y(i,j,k) = \sum_{l=1}^{M} A_l^l F1_i^l F2_j^l F3_k^l + \epsilon_{(i,j,k)}, \tag{4.5}$$

where $Y(i,j,k)$ is one observation of three-dimensional experimental data matrix Y with size (I,J,K) and $i = 1,2...I$; $j = 1,2...J$; $k = 1,2...K$. M is the number of components. $F1^m$, $F2^m$, and $F3^m$ are one-dimensional functions and called shapes while $\epsilon_{(i,j,k)}$ is the error. Orekhov et al. (2001) developed MUNIN based on this three-way decomposition. Although their method is widely used for NMR data analysis, amplitudes of signals at various dimensions may not be of comparable size (Luan et al., 2005). The reason could be that amplitude differences between components tend to increase to some extend after a series of iteration so that they are not comparable. Luan et al. (2005) then improved their method by introducing L_2 penalty term to the least-square minimization. The new procedure can be written as follows.

$$MIN_{F1,F2,F3,A,M} \sum_{t=1}^{R} (\sum_{i,j,k} (Y(i,j,k) - \sum_{l=1}^{M} A_l^l F1_i^l F2_j^l F3_k^l))^2 + \lambda \sum_{l=1}^{M} (A_l^l)^2, \tag{4.6}$$

where λ is the tuning parameter. The algorithm used to solve this high-dimensional minimization problem is PARAFAC provided by Harshman and Lundy (1984). This method is efficient in terms of amplitudes estimation and resolve overlapped signals. As comparing ridge regression to ordinary least squares approach, this method is better than MUNIN in terms of accuracy of the estimation. However, the L_2 penalty term in the objective function (4.6) may not be efficient in terms of components selection whereas the ratio of signal and noise is small. Since the penalty can be adjusted based on the components left after each selection step. Also the penalty term doesn't do variable selection. That is, a statistical criterion on selecting a correct M is still needed.

We thus propose a new penalized minimum least square method so that the proposed method not only has accurate estimates as Luan et al's method does but also selecting the correct M. Therefore, based on the result of Qiu, Yin and Wang (2008), instead of L_2 penalty,

we propose a general adaptive penalty term to equation (4.6). That is,

$$MIN_{F1,F2,F3,A,M} \sum_{t=1}^{R}(\sum_{i,j,k}(Y(i,j,k) - \sum_{l=1}^{M} A_l^l F1_i^l F2_j^l F3_k^l))^2 + A^T \lambda \Lambda W A. \tag{4.7}$$

Here $W = \text{diag}(W_1, \cdots, W_M)$ is a diagonal weight matrix with lth diagonal element $W_l = \frac{1}{|A_l^l|^\gamma}$ with γ being a pre-chosen value, $\lambda \geq 0$ is the tuning parameter and $\Lambda = \text{diag}(\lambda_1, \cdots, \lambda_M)$ is a diagonal matrix with positive elements such that $\sum_{l=1}^{M} \frac{1}{\lambda_l} W_l = M$. This adaptive penalty term can improve the efficiency of components selection. The weight matrix can be adjusted based on selected components so that component selection procedure becomes much more efficient. The parameter matrix Λ is also updated during the process. Once one component is deleted, it will never enter again. In this way, this method is very efficient of components selection.

Note that if $\lambda = 0$, then this method is reduced to MUNIN, a TWD method; if $W = I_M$ is fixed as an identity matrix, then this method is reduced to the approach provided by Luan et al. (2005). On the other hand, demonstrated by Qiu, Yin and Wang (2008), even if it appears a general adaptive ridge, it is in fact equivalent to a combination of L_1 and L_2 penalty terms for some nonnegative parameters λ_0^* and λ_1^* as follows:

$$MIN_{F1,F2,F3,A,M} \sum_{t=1}^{R}(\sum_{i,j,k}(Y(i,j,k) - \sum_{l=1}^{M} A_l^l F1_i^l F2_j^l F3_k^l))^2$$
$$+\lambda_0^* A^T W A + \lambda_1^* \sum_{l=1}^{M} W_l |\mathbf{A}_l^l|.$$

Thus with L_1 penalty, this method can effectively select variables, achieving the estimation of M; while with L_2 penalty, the method can improve the accuracy. Finally, a statistical criterion of BIC is used to select the tuning parameter λ.

4.3 Algorithm for ATWD

As discussed previously, the limitation of minimum least square method is that the parameter estimation is not reliable sometime. Noises and true signals cannot be distinguished from

each other. The method developed by Luan et al. (2005) enhanced the MUNIN so that it can pick true signals out and delete noises even when the ratio of noises and signals is relatively big. ATWD including a more general adaptive data-driven parameters shall improve both aforementioned methods by further estimating the correct number of components. Below is our algorithm.

1. Guess number of component M (Typically, to be conservative big M is used) and set W (where initial A_l^l's are from TWD with γ being fixed number of 0, .5 or 1 based on Qiu, Yin and Wang (2008)). Set $\tilde{\Lambda} = \lambda\Lambda$ with diagonal elements equal to 1, from which an initial λ is obtained under the constraint on Λ.

2. Reduce the array Y to a matrix S by concatenation of the second and third dimensions.

3. Find initial $F1^{(s)}$ which is constructed from upper largest singular vector of matrix S. Here s is step index.

4. Construct matrix $D(l)_{Jxk}^{(s)} = \sum_i Y_{i,j,k} F1(l)_i^{(s)}$.

5. Obtain $A^{(s)}$, $F_2^{(s)}$ and $F_3^{(s)}$ by single value decomposition of matrix $D(l)^{(s)}$. They are corresponding to largest singular value, the left and right singular vectors.

6. Update each shape matrix by fixing other two shape matrices via solving a least square problem with quadratic penalty terms (Orekhov et al., 2001).

7. Update diagonal matrix $A^{(s)}$ by normalizing each shape matrix.

8. For $j = 1, \cdots, M_s$ (where for the initial step $M_s = M$), calculate $a_l^{(s)} = \frac{|A_l^{l(s)}|}{\sum W_l^{(s)} |A_l^{l(s)}|}$. If $a_l^{(s)} < C$, delete lth component; if $a_l^{(s)} \geq C$, keep it. Here C is selection criteria.

9. Update $W^{(s+1)}$ with TWD method for the remaining components. Update $\Lambda^{(s+1)}$ with the remaining components by $\lambda_l^{(s+1)} = (M|A_l^{l(s)}|/[\sum_{l=1}^{M_{s+1}} W_l^{(s+1)} |A_l^{l(s)}|])^{-1}$.

10. Update $A^{(s+1)}$ by going through steps 2 to 7. If $\max |A_l^{l(s+1)} - A_l^{l(s)}|/[1 + A_l^{l(s+1)}]$ is less than 10^{-6}, then it is converged; otherwise go to step 8.

Parameter C and the tuning parameter λ affect selection results. The range of C is from 0 to 1 and the range of λ is from 0 to 200. The best C and λ values are determined by BIC criterion for sample size n which can be described as follows.

$$BIC = n * ln(\frac{(Y - \hat{Y}(C,\lambda))^T(Y - \hat{Y}(C,\lambda))}{n}) + M(C,\lambda) * ln(n), \tag{4.8}$$

where to emphasize the dependence on C and λ, $\hat{Y}(C,\lambda)$ is the estimated values of Y at C and λ and $M(C,\lambda)$ is the estimated M at C and λ, when convergence is reached. Figures 4.1 and 4.2 show typical BIC plots for relaxation studies and 3D NOESYs data analysis.

4.4 Simulation studies

In this section, we want to test the efficiency of this algorithm in handling the influence of noise and peak overlap. The accuracy and precision of estimated parameters such as relaxation times will be evaluated with or without adaptive penalization. Although this algorithm is robust in terms of line shapes, the ideal shapes were used to simulate input spectrum. In this way, peak overlap can be controlled easily so that the accuracy of estimated parameters can be determined. Following Luan et al. (2005), it is assumed that shapes in two frequency domains are absorption Lorentzian line-shapes:

$$f(\Omega) = W\frac{\alpha}{1 + \alpha^2(\Omega - \Omega_0)^2}, \tag{4.9}$$

where line width is determined by the inverse of α which is nonnegative, the center of each peak corresponds to the value of Ω_0 and W is a normalization parameter such that integration of equation (4.9) is unity. Further, we assume the time dimension can be described by an exponential function with normalization factor A and relaxation time T^0:

$$g(t) = Aexp(-t/T^0). \tag{4.10}$$

In each frequency dimension, 40 points obtained by applying equation (4.9) were used to construct the frequency planes. In each frequency dimension, there were two peaks and they are located on the diagonal of the frequency plane. The initial intensity (i.e., amplitude) of

strong peak relative to weak peak is 3:2. The value of Ω_0 is adjusted to control the overlap of two peaks. The values of Ω_0's for two peaks were set at (20,12), (18,12), (18,14), (15,15) and the value of α is 1/4. In the time dimension, the values of t were set to be 0, 1, 2, 3, 5, 8, 13, 18, 25, 32, 40, 50, 60, 70 ms. Then, an input data with 14 planes were simulated with equation (4.5), equation (4.9) and equation (4.10). Furthermore, the relaxation time Ω_0 of two peaks were set at 30 and 50ms. The signal noise ratio was defined as the ratio of amplitudes for signals and noises. The values of the signal noise ratio are set at 10, 30, 50, 70 and 90. As frequency planes, a pool of 600 frequency planes was constructed. Then 14 frequency planes were randomly selected for each of 50 repeated runs. The ATWD algorithm was applied to various simulated spectra corresponding to different combination of signal overlap (i.e., 4 pairs shown before) and signal noise ratio (i.e., 5 values shown before). Relaxation times (T) were estimated from ATWD output by a least square fit of the output. Then the accuracy and precision of the 50 estimated parameters for each combined conditions was calculated with the following formula:

$$log(acc) = log\sqrt{\frac{\sum(T_i - T^0)^2}{50(T^0)^2}}, \tag{4.11}$$

where T^0 is the true value of T in equation (4.10), T_i is the estimated values of T at ith time ($i = 1, \cdots, 50.$) and acc is an abbreviation of $accuracy$.

$$log(prec) = log(\frac{var_i}{T^0}), \tag{4.12}$$

where var_i is the estimated values of variance at ith time for errors and $prec$ is an abbreviation of $precision$.

4.5 NMR spectroscopy

The 3D HNCO data was acquired on a NESG target from Pyrococcus furious, PF0385 on Varian Inova600. 900, 32, and 30 complex points were collected along 1H, 13C and 15N

dimensions with the acquisition time 90, 15.2 and 16.7 ms, respectively. Two scans were collected for each fid with a recycle delay 1s. A 13C and 15N labeled protein sample at 1.5 mM in 20 mM TRIS-MOPS, 50 mM KCl, 5 mM DTT and pH 7.0 was employed for the data collection. The spectral widths for the three dimensions were 10, 2.11 and 1.80 kHz. More detail information about this data can be obtained from the authors.

Only 640 out of 900 one-dimensional 1H, which are amide proton chemical shift region, was Fourier transformed one by one using NMRPipe software with square sine-bell weighting functions. In order to reduce data and number of components, cubic regions (64*60*100) were extracted by taking points from 300 to 399 (33.33pm to 44.33ppm) in 1H dimension. The position of this region was corresponding to a crowded region of the spectrum along hydrogen dimension including 5-8 components. The exact number of components can not be determined due to signal overlap. Note that only 1H dimension was Fourier transformed, while the dimensions 13C and 15N remained in time domain. After ATWD calculation for spectrum reconstruction, 1D Fourier transform in the 13C and 15N dimensions were performed.

To sample the data set irregularly, 75 out of 100 planes were selected and removed randomly. The remaining planes (3, 5, 8, 11, 14, 17, 20, 23, 25, 28, 31, 33, 36, 39, 42, 45, 47, 50, 53, 58, 60, 63, 87, 93, 98; odd and even numbers represents the real and imaginary planes), representing 25% of the data, were collected and referred as the reduced data set that was actually used in the section 6.2.

4.6 Results

4.6.1 Simulation studies

As described in section 4.4, input spectra consisting of 14 planes were constructed via varying the two parameter values (i.e., signal overlap and signal noise ratio). Overlaps between two peaks were changed from separation to total overlap. The extent of overlaps between two peaks for four simulations of relaxation data are displayed in Fig. 4.3. From Fig. 4.3.A.

to Fig. 4.3.D., separation in each dimension of the peak centers is 8, 6, 4, 0 spectral points respectively. In this figure, only first plane ($t = 0$) was displayed. Logarithms of average measurement accuracy as a function of signal noise ratio under calculation conditions without (thick solid line) and with penalty (thin dash line) were reported in Fig. 4.4 (weight parameter $\gamma = 0$) and Fig. 4.5 (weight parameter $\gamma = 1$). These two figures display similar patterns. That is, with the extent of overlap between two peaks increasing, improvement of accuracy calculated with penalty increases. In addition, accuracy for strong peak is a little bit higher than weak peak. Based on these results, weights do not affect parameter estimation significantly. Fig. 4.6 (weight parameter $\gamma = 0$) and Fig. 4.7 (weight parameter $\gamma = 1$) reported logarithms of average measurement precisions as a function of signal noise ratio under calculation conditions without (thick solid line) and with penalty (thin dash line). Precisions follow similar pattern of accuracy in terms of different peak overlap, signal noise ratio and amplitude of peaks. For each simulation, a series of number of components as initial M (i.e., 2, 5, 10, 15) are considered and 5 was actually used in later simulations. The results with equal weights (i.e., $\gamma = 0$) indicate that initial number of components M more than 2, the estimated values of M end up in 2 with 70 out of 100 times of simulations. The left 30 estimated values of M are more than 2. The results with unequal weights (i.e., $\gamma = 1$) indicate that initial number of components M more than 2, the estimated values of M end up in 2 with 71 out of 100 times of simulations. The left 29 estimated values of M are more than 2. Weights seem not affect component selection significantly. In all, ATWD method tends to overestimate the number of components, but never misses components.

4.6.2 Decomposition of a 3D HNCO spectrum

Using the ATWD procedure, a spectral region of a 3D HNCO spectrum, as described in the previous section, was decomposed into 20 ($= \hat{M}$) components with bigger initial guesses (M=30). Fig. 4.8 (weight parameter $\gamma = 0$) and Fig. 4.9 (weight parameter $\gamma = 1$) show the 1H-dimensional shapes of the two components from raw experimental data. These two

figures are very similar to each other. Therefore, weights do not make big impact on spectrum reconstruction. The 1H dimension has been Fourier transformed before the spectrum was processed by ATWD. 1H projections from the 3D HNCO spectrum reconstructed with reduced data (25% of raw data) are shown in Fig. 4.10 (weight parameter $\gamma = 0$) and Fig. 4.11 (weight parameter $\gamma = 1$). Clearly, these two figures are very similar to each other. Interestingly, the spectrum reconstructed from the ATWD with reduced data are very similar to the spectrum reconstructed with full data.

We also randomly select 25% of the raw data for 100 times to produce 100 different reduced data. For each reduced data, a series of number of components as initial M (i.e., 20, 30, 40, 50) is used. The results with equal weights ($\gamma = 0$) and unequal weights ($\gamma = 1$) indicate that using initial number of components $M = 30$, the estimated values of M end up in 20 ($= \hat{M}$) with 77 and 80 out of 100, respectively. The remaining 23 or 20 estimated values of M are more than 20. Therefore, ATWD method tends to overestimate the number of components but again never misses the components.

4.7 Conclusion

In this paper, we proposed a new method of ATWD which includes both MUNIN (A TWD approach) and the methodprovided by Luan et al. (2005) as its special cases, thus it inherits advantages of both methods. In addition, ATWD can statistically estimate the number of components in the decomposition. The method has been applied to simulated relaxation data and real data. The accuracy and precision of ATWD method have been tested based on simulated relaxation data. The results indicate its usefulness not only in accuracy and precision but also in providing a good estimate of the number of components. The 3D HNCO data was used to test the efficiency of ATWD for spectrum reconstruction. In conclusion, ATWD method is an efficient method for spectrum reconstruction. Extension of ATWD to NMR data with dimensions more than 3 may be possible.

4.8 References

[1] Alan S. Stern, Kuo-Bin Li, and Jeffrey C. Hoch (2002) Modern spectrum analysis in multidimensional NMR spectroscopy. Comparison of linear-prediction extrapolation and maximum-entropy reconstruction , *J. Am. Chem. Soc.* 124, 1982-1993.

[2] Carroll, J.D. and Chang, J. (1970) *Psychometrica* 35, 283-319.

[3] Carroll, J.D. and pruzansky, S. (1984) In Research Methods for Multimode Data Analysis, Praeger, New York, NY, pp.372-402.

[4] Harshman, R.A. (1970) *UCLA Working Paper in Phonetics* 16, 1-84.

[5] Harshman, R.A. and Lundy, M.E. (1984) In Research Methods for Multimode Data Analysis, Praeger, New York, NY, pp.122-215.

[6] Hoch, J.C. and Stern, A.S. (1996) In Encyclopedia od Nuclear Magnetic Resonance, John Wiley, London, pp. 2980-2988.

[7] Koehl, P. (1999) *Prog. NMR Spectrosc.* 34, 257-299.

[8] Luan, T., Orekhov, V.Y., Gutmanas, A., Billeter, M. (2005) Accuracy and robustness of three-way decomposition applied to NMR data, *J. Magn. Reson.* 174, 188199.

[9] Malmodin, D. and Billeter, M. (2005) High-throughout analysis of protein NMR spectra. *Progress in Nuclear Magnetic Resonance Spectroscopy* 46, 109-129.

[10] Orekhov, V.Y., Ibraghimov, I.V. and Billeter, M. (2001) Optimizing Resolution in Multidimensional NMR by Three-way Decomposition. *J. Biomol.NMR* 27, 165-173.

[11] Qiu, J., Yin, X., and Wang, H. (2008) Generalized adaptive ridge: a data driven approach. *Submitted to Technomics*

[12] Schmieder, p., Stern, A.S., Wagner, G., Hoch, J.C. (1997) *J. Magn. NMR* 125, 332-337.

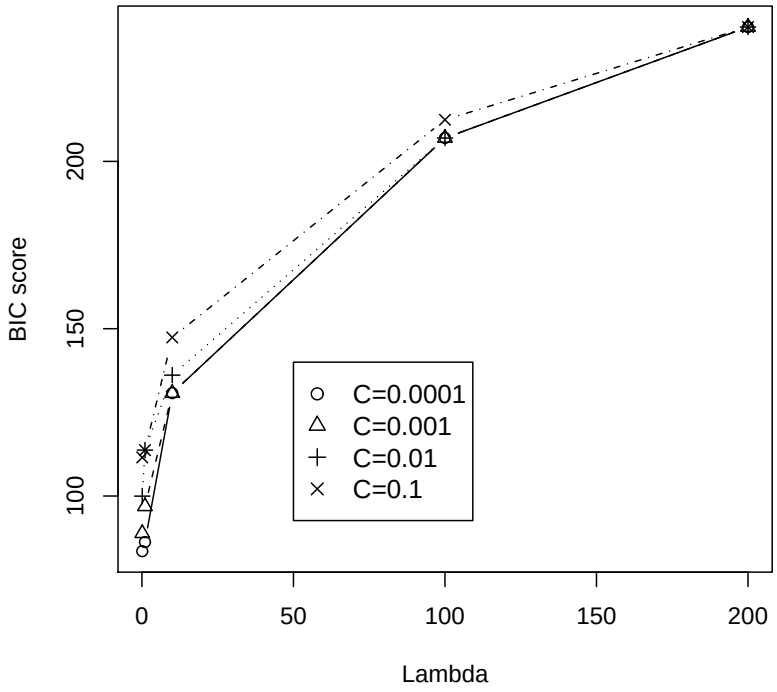

Figure 4.1: Tuning parameter selection (BIC) for relaxation simulation studies.

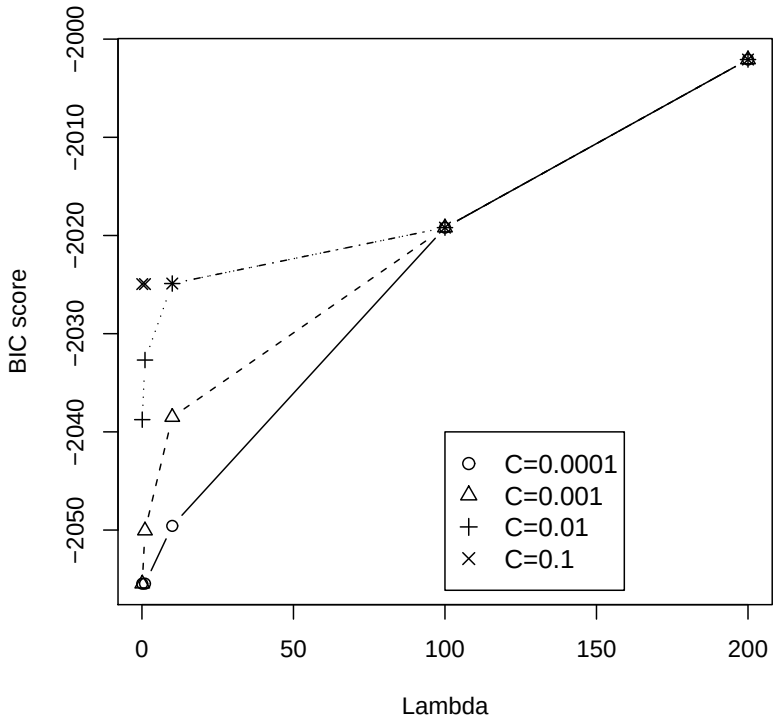

Figure 4.2: Tuning parameter selection (BIC) for 3D NOESYs studies.

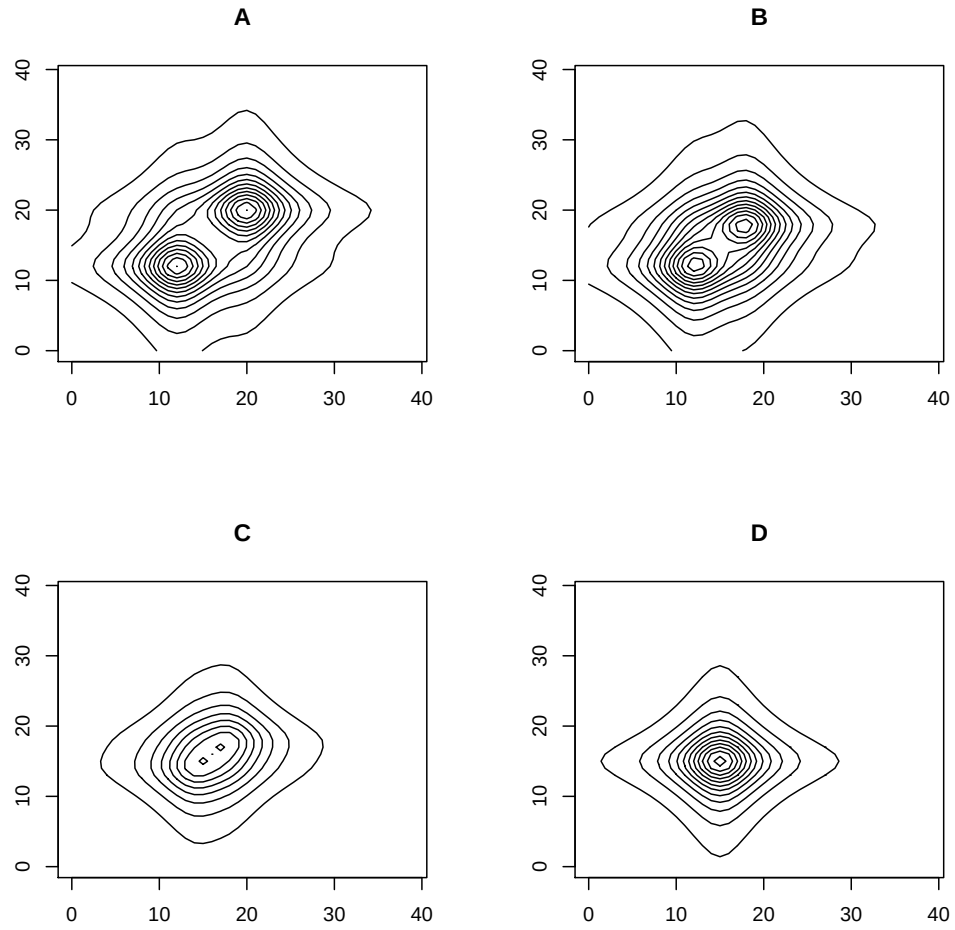

Figure 4.3: The extent of overlap of two peaks for four different simulations of relaxation data sets when $t = 0$. Two axis represent the centers of peaks (Ω_0). From A to D, the values of Ω_0's for two peaks are set at (20,12), (18,12), (18,14), (15,15).

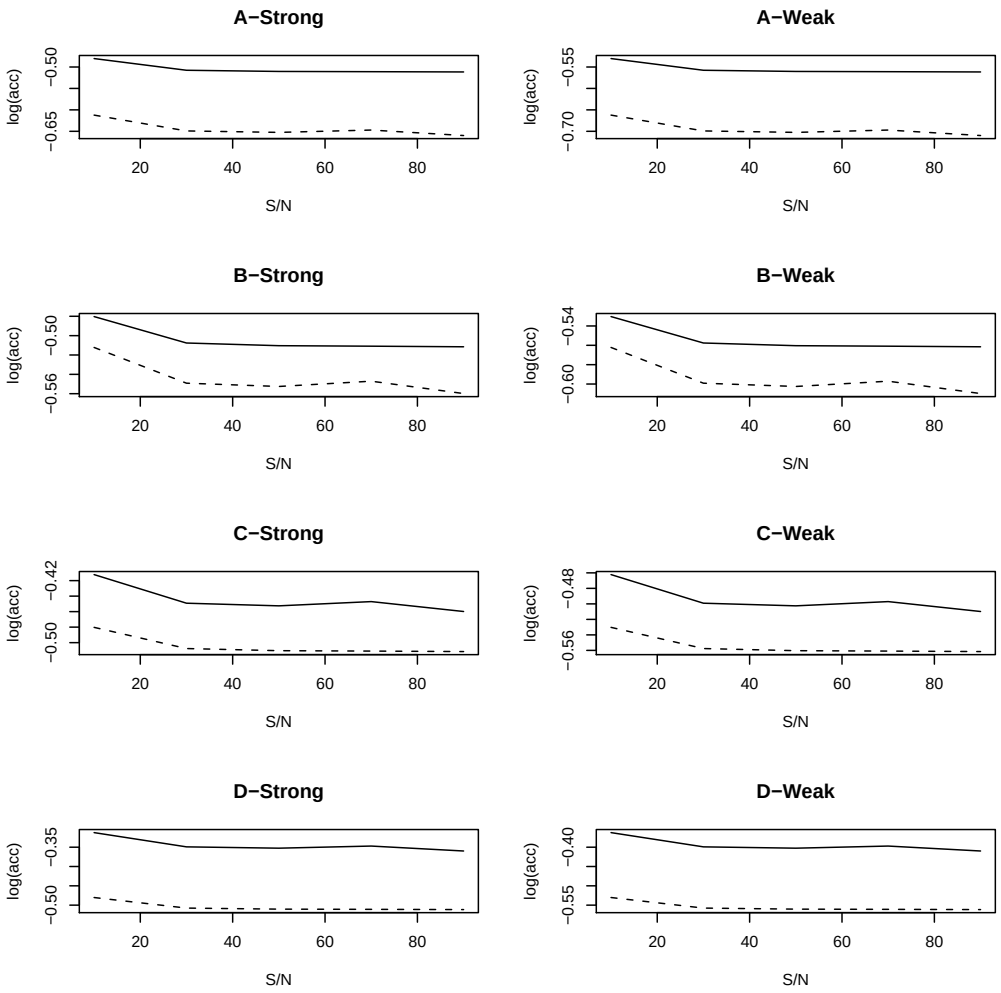

Figure 4.4: Accuracy as a function of signal noise ratio for various overlap conditions. The panels on the left side labeled "strong" correspond to results for the strong peak in the overlap situations A-D. The panels on the right side, labeled "weak" correspond to weal peak in the overlap situations A-D. The amplitude ratio between strong and weak peaks are 3:2. The dash and solid lines correspond to ATWD calculations with penalty term ($\gamma = 0$) and without penalty.

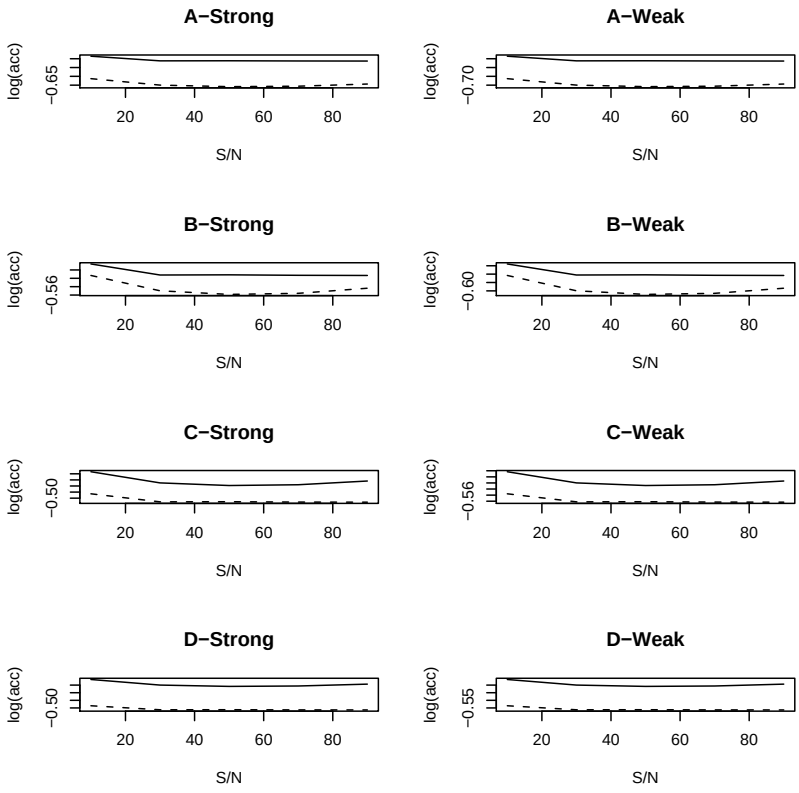

Figure 4.5: Accuracy as a function of signal noise ratio for various overlap conditions. The panels on the left side labeled "strong" correspond to results for the strong peak in the overlap situations A-D. The panels on the right side, labeled "weak" correspond to weal peak in the overlap situations A-D. The amplitude ratio between strong and weak peaks are 3:2. The dash and solid lines correspond to ATWD calculations with penalty term ($\gamma = 1$) and without penalty term.

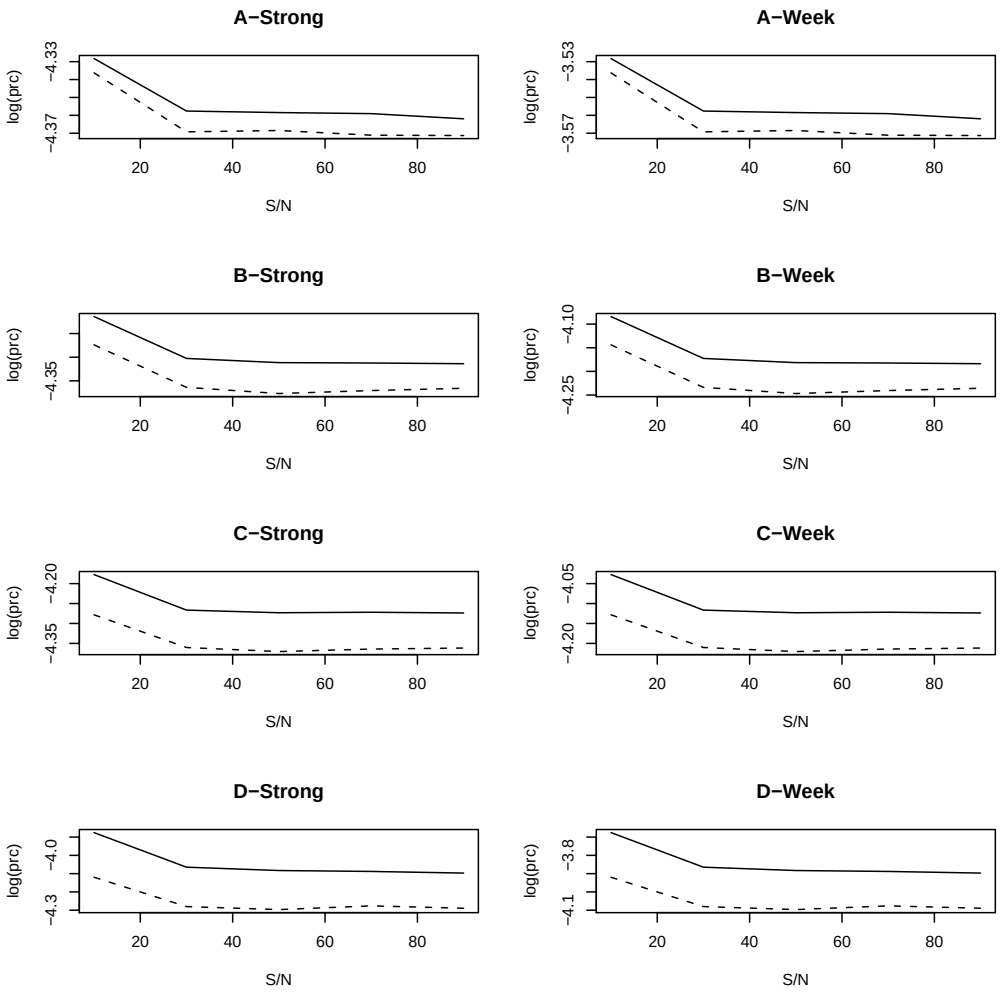

Figure 4.6: Precision as a function of signal noise ratio for various overlap conditions. The panels on the left side labeled "strong" correspond to results for the strong peak in the overlap situations A-D. The panels on the right side, labeled "weak" correspond to weal peak in the overlap situations A-D. The amplitude ratio between strong and weak peaks are 3:2. The dash and solid lines correspond to ATWD calculations with penalty term ($\gamma = 0$) and without penalty term.

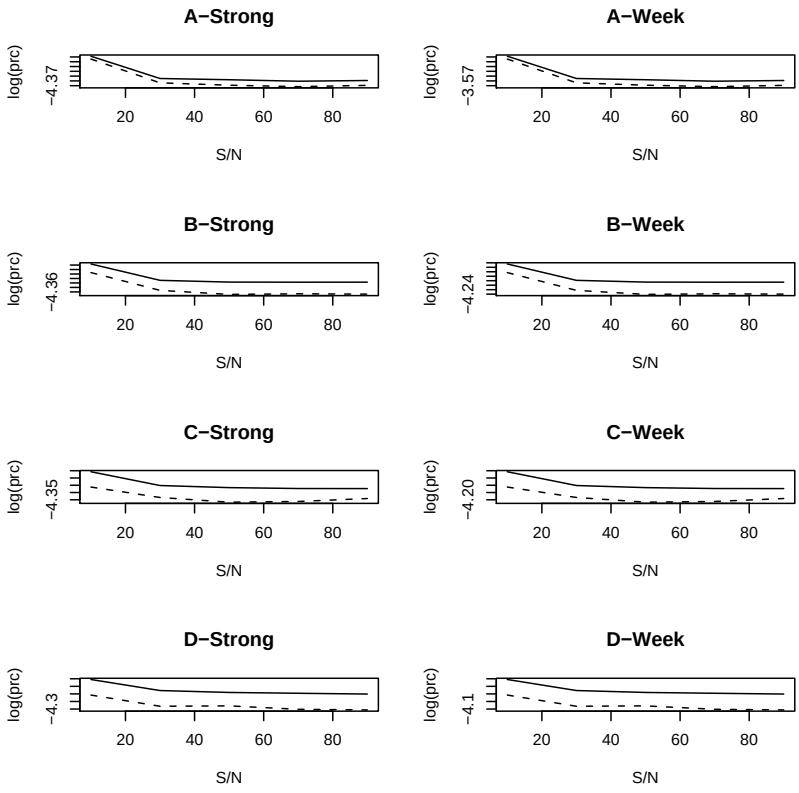

Figure 4.7: Precision as a function of signal noise ratio for various overlap conditions. The panels on the left side labeled "strong" correspond to results for the strong peak in the overlap situations A-D. The panels on the right side, labeled "weak" correspond to weal peak in the overlap situations A-D. The amplitude ratio between strong and weak peaks are 3:2. The dash and solid lines correspond to ATWD calculations with penalty term ($\gamma = 1$) and without penalty term.

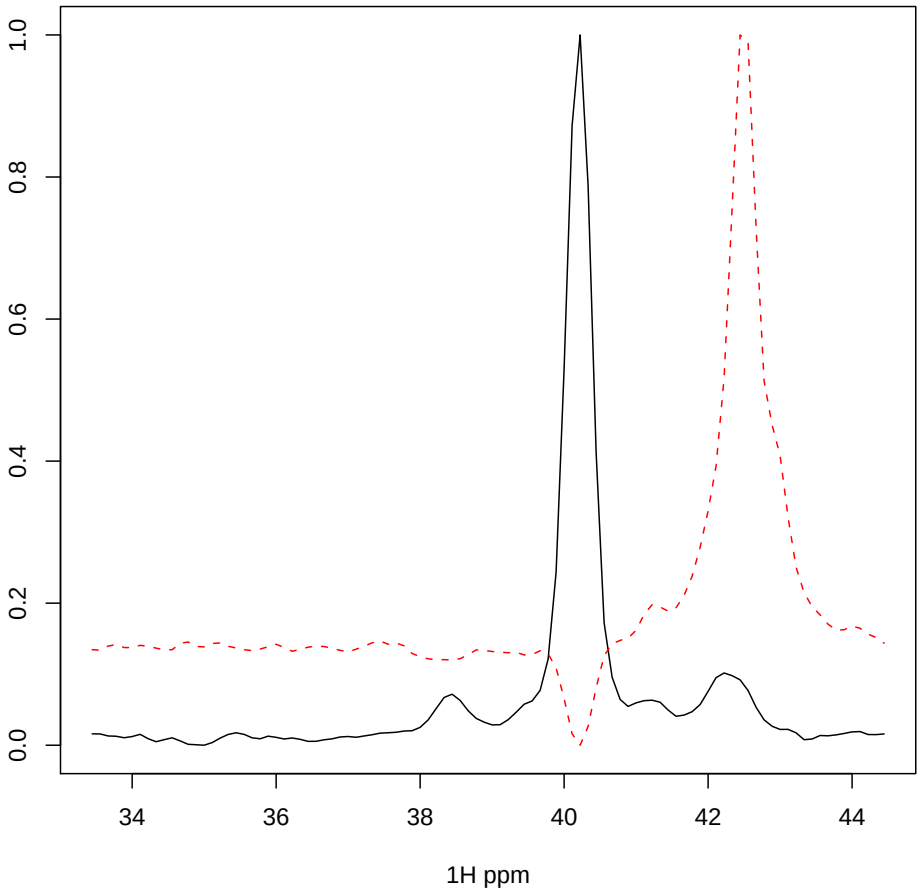

Figure 4.8: Normalized one-dimensional shapes of 2 (out of 20) estimated components reconstructed by ATWD using $\gamma = 0$ with raw data are referred to as component 1 (heavy lines) and component 2 (thin lines).

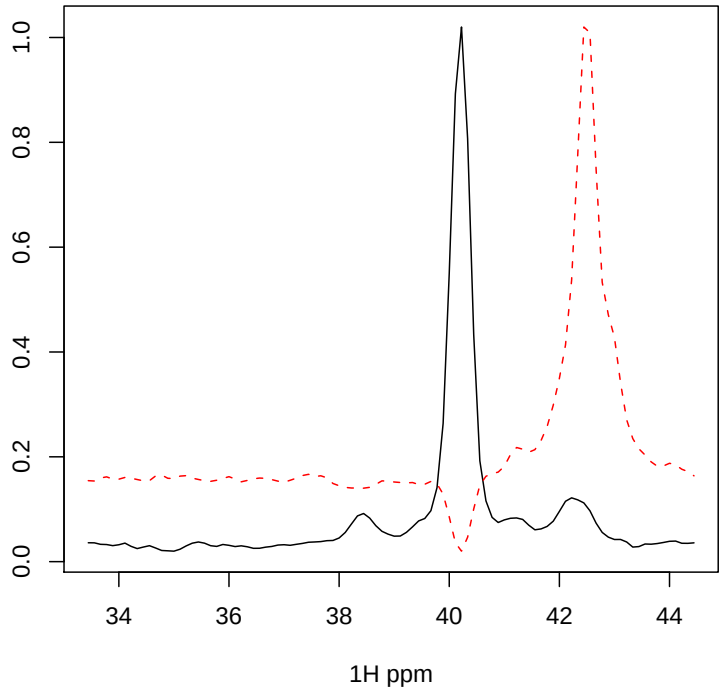

Figure 4.9: Normalized one-dimensional shapes of 2 (out of 20) estimated components reconstructed by ATWD using $\gamma = 1$ with raw data are referred to as component 1 (heavy lines) and component 2 (thin lines).

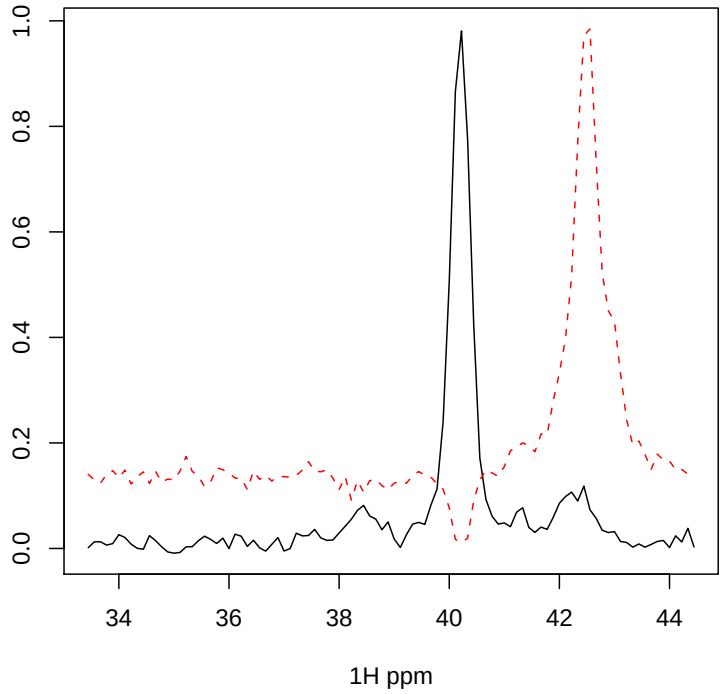

Figure 4.10: Normalized one-dimensional shapes of 2 (out of 20) estimated components reconstructed by ATWD using $\gamma = 0$ with reduced data are referred to as component 1 (heavy lines) and component 2 (thin lines).

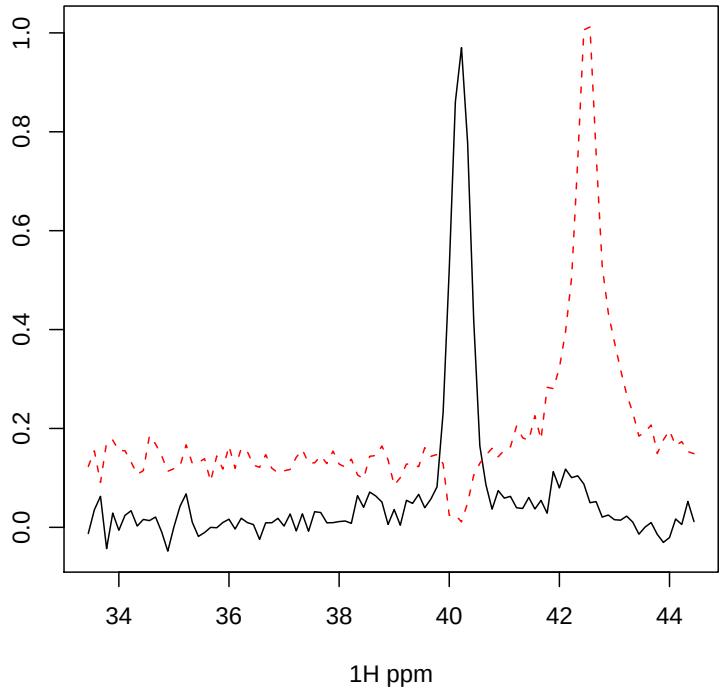

Figure 4.11: Normalized one-dimensional shapes of 2 (out of 20) estimated components reconstructed by ATWD using $\gamma = 1$ with reduced data are referred to as component 1 (heavy lines) and component 2 (thin lines).

Bibliography

[1] Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. *Second International Symposium on Information Theory*, eds. Petrov, B.N., and Csaki, F., 261-281.

[2] Alan S. Stern, Kuo-Bin Li, and Jeffrey C. Hoch (2002) Modern spectrum analysis in multidimensional NMR spectroscopy. Comparison of linear-prediction extrapolation and maximum-entropy reconstruction , *J. Am. Chem. Soc.* 124, 1982-1993.

[3] Carroll, J.D. and Chang, J. (1970) *Psychometrica* 35, 283-319.

[4] Carroll, J.D. and pruzansky, S. (1984) In Research Methods for Multimode Data Analysis, Praeger, New York, NY, pp.372-402.

[5] Cessie, S. and Houwelingen J.C. (1992) Ridge estimators in logistic regression.*Appl. Statist.* 41, 191-201.

[6] Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression. *Ann. Statist.* 32, 407499.

[7] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* 96, 1348–1360.

[8] Fu, W. J. (1998) Penalized Regression: The Bridge Versus the LASSO. *Journal of Computational and Graphical Statistics* 7, 397-416.

[9] Grandvalet, Y. (1998). Least absolute shrinkage is equivalent to quadratic penalization. In *L. Niklasson, M. Boden, and T. Ziemske, editors, ICANN98* pages 201-206. Springer.

[10] Golub et al. (1999) Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science* 286, 531538.

[11] Harshman, R.A. (1970) *UCLA Working Paper in Phonetics* 16, 1-84.

[12] Harshman, R.A. and Lundy, M.E. (1984) In Research Methods for Multimode Data Analysis, Praeger, New York, NY, pp.122-215.

[13] Hawkins, D. and Yin, X. (2002). A faster algorithm for ridge regression for reduced rank data. *Journal of Computational Statistics & Data Analysis* 40, 253–262.

[14] Hoch, J.C. and Stern, A.S. (1996) In Encyclopedia od Nuclear Magnetic Resonance, John Wiley, London, pp. 2980-2988.

[15] Hoerl, A.E. and Kennard, R.W. (1970a). Ridge regression: applications to nonorthogonal problems. *Technometrics* 12, 69–82.

[16] Hoerl, A.E. and Kennard, R.W. (1970b). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* 12, 55–67.

[17] Koehl, P. (1999) *Prog. NMR Spectrosc.* 34, 257-299.

[18] Leng, C., Lin, Y. and Wahba, G. (2006). A note on lasso and related procedures in model selection. *Statistica Sinica* 16, 1273–1284.

[19] Lu, W. and Zhang, H. (2007). Variable selection for proportional odds model. *Statistics in Medicine* 26, 3771-3781.

[20] Luan, T., Orekhov, V.Y., Gutmanas, A., Billeter, M. (2005) Accuracy and robustness of three-way decomposition applied to NMR data, *J. Magn. Reson.* 174, 188199.

[21] Malmodin, D. and Billeter, M. (2005) High-throughout analysis of protein NMR spectra. *Progress in Nuclear Magnetic Resonance Spectroscopy* 46, 109-129.

[22] Orekhov, V.Y., Ibraghimov, I.V. and Billeter, M. (2001) Optimizing Resolution in Multidimensional NMR by Three-way Decomposition. *J. Biomol.NMR* 27, 165-173.

[23] Park, M. Y. and Hastie, T. (2007) An L1 regularization-path algorithm for generalized linear models. *J. R. Statist. Soc. B* 69, 659-677.

[24] PLATT, J. (1998) Sequential minimal optimization: A fast algorithm for training support vector machines. *Technical Report* MSR-TR-98-14, Microsoft Research.

[25] Qiu, J., Yin, X., and Wang, H. (2008) Generalized adaptive ridge: a data driven approach. *Submitted to Technomics*

[26] Rosset, S. and Zhu, J. (2007). Piecewise linear regularized solution paths. *Annals of Statistics*, 35, 1012-1030.

[27] Schmieder, p., Stern, A.S., Wagner, G., Hoch, J.C. (1997) *J. Magn. NMR* 125, 332-337.

[28] Schwarz, G. (1979). Estimating the dimension of a model. *Annals of Statistics*, 6, 461-464.

[29] Shi, P., and Tsai, C.-L. (2002). Regression model selection – a residual likelihood approach. *Journal of Royal Statistical Society, Series B.* 64, 237-252.

[30] Stephenson, D. S. (1988) *Prog. NMR Spectrosc* 20, 515-626.

[31] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B* 58, 267-288.

[32] Tibshirani, R. (1997). The lasso method for variable selection in the Cox model. *Statistics in Medicine* 16, 385395.

[33] Tibshirani, R. Saunders, M. Rosset, S. Zhu, J. and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *J. R. Statist. Soc. B* 67, 91108.

[34] Turlach, B. A. (2006). An even faster algorithm for ridge regression of reduced rank data. *Journal of Computational Statistics & Data Analysis*, 50, 642–658.

[35] Wang, H. and Leng, C. (2007). Unified LASSO estimation by least Squares Approximation. *Journal of the American Statistical Association*, 102, 1039-1048.

[36] Wang, H., Li, G. and Jiang, G. (2007). Robust regression shrinkage and consistent variable selection via the lad-lasso. *Journal of Business and Economics Statistics* 25, 347–355.

[37] Wang, H., Li, G. and Tsai, C. L. (2007). Regression coefficient and autoregressive order shrinkage and selection via lasso. *Journal of Royal Statistical Society, Series B* 69, 63–78.

[38] Wang, H., Li, R. and Tsai, C .L. (2007). Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika* 94, 553–568.

[39] Wood, S. N. (2000). Modeling and Smoothing Parameter Estimation with Multiple Quadratic Penalties. *J. R. Statist. Soc. B* 62, 413-428

[40] Yuan, M. and Lin, Y. (2007). On the nonnegative garrote estimator. *Journal of the Royal Statistical Society, Series B* 69, To appear.

[41] Zhang, H. and Lu, W. (2007). Adaptive Lasso for COX's Proportional Hazards Model. *Biometrika* 1-17.

[42] Zhao, P. and Yu, B. (2007). On model selection consistency of lasso. *Journal of Machine Learning Research* 7, 2541–2567.

[43] Zhu, J. and Hastie, T. (2004). Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 5, 427-443.

[44] Zou, H. (2006). The adaptive-LASSO and its oracle properties. *Journal of American Statistical Association* 101, 1418-1429.

[45] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. R. Stat. Soc. B* 67, 301-320.

[46] Zou, H., Hastie, T. and Tibshirani, R. (2007). On the 'degrees of freedom' of lasso. *The Annals of Statistics*, 35, 2173-2192.