The Pennsylvania State University
The Graduate School
College of Engineering

SPARSE RECOVERY OF TIME-VARING STREAMING DATA USING HOMOTOPY METHOD

A Thesis in
Industrial Engineering
by
Xue Wang

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2015

The thesis of Xue Wang was reviewed and approved* by the following:

Tao Yao
Professor of Industrial Engineering
Thesis Advisor

Lingzhou Xue
Professor of Statistics
Reader

Harriet Black Nembhard
Head of the Harold and Inge Marcus Department of Industrial and Manufacturing Engineering
Department Head

*Signatures are on file in the Graduate School.

Abstract

Many existing algorithms for regularized least square regression assumes that the true parameters to be stable and not change with time. However, the algorithm and framework for recovering time-varying signal with online updating is not well researched. Some people proposed the kind of homotopy $l1-$ minimization methods for dealing with online updating data, which has been shown to work well, but there are also some limitations: (1) the result of $l1-$ minimization method is biased; (2) the homotopy algorithm has been proved to be a kind of exponential method. In some special cases, the homotopy $l1-$ minimization method may work bad. In this thesis, we constructs a novel homotopy algorithm for the situation of time-varying signal with online updating. In the algorithm: (1) we fold concave regularation instead of $l1$r-egularation, such as SCAD, which has been proved to have better statistical properties; (2) the complexity bound of our method is $O(N^2/\sqrt{\epsilon})$. Besides the complexity bound, our method also has a special property. No matter how dramatically the significant parameters changes, our method will converge in very few steps if the significant parameters remain significant and insignificant parameters remain insignificant. In the numerical experiments, we present our method's performance compared to some other methods in several different situation, such as, urban traffic travel time and image recovery.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Problem Statement

In this paper, we discuss a problem of recovering a time varying sparse system from incomplete streaming data, which has been found its applications in many fields, such as, signal processing[3, 11, 42], dynamic traffic flow estimation[23], detecting moving objects in dynamic scenes[47] and shallow-water acoustic communication[28]. Most of traditional sparse recovery methods assume the signal being stable. In our analysis, we release this assumption to the signal follows a linear Gauss-Markov model, which is:

$$\beta_{t+1} = A_t\beta_t + v_t \quad t = 0, 1, 2, ... \tag{1.1}$$

Where β_t is the signal at time t, A_t is a prediction matrix at time t that couples β_t and β_{t+1} and v_t is iid Gaussian random variable that follows $N(0, \sigma_v^2)$. If we set $\sigma_v = 0$, $A_t = I$, the equation(1.1) will degenerate to $\beta_{t+1} = \beta_t$, which is just the situation of stable signal.

We also assume the unknown signal β_t is a vector with finite elements and fits the following linear system:

$$y_t = X_t\beta_t + \epsilon_t \quad t = 0, 1, 2, ... \tag{1.2}$$

where y_t is the vector of responses at time t, $X_t = \{x_{1,t}^T, x_{2,t}^T, ..., x_{n,t}^T\}^T$ is the measurement matrix at time t, where $x_{i,t},\ i = 1, 2, ..., n$ is the ith measurement

corresponding to $y_{i,t}$ and ϵ_t $N(0, \sigma_\epsilon^2)$ is a vector of error which follows iid normal distribution. We can used ordinary least squares regression(OLS) or Kalman Filter(KF) to get an optimal estimation[39].

However, both OLS and KF have some defects: (1) $X_t^T X_t$ should be well conditioned; (2) the length of β_t can't be larger than the $rank(X_t)$[7] and (3) the solution could be instable when some predictors are highly correlated[21].

To overcome these three difficulties, people developed the regularized least square regression [16, 22, 40, 49] . The formulation of regularized least square regression is:

$$\beta = \arg_\beta \min \frac{1}{2}\|y - X\beta\|_2^2 + P_\lambda(|\beta|) \tag{1.3}$$

where $P_\lambda(.)$ is the regularation function. However, these methods are not suitable for dealing the situation of streaming updating and unknown signal changing with time. Since can not collect all measurements or measuring the entire signal all at a time, these tasks need to be finished sequentially. We build the model with time varying observations as:

$$y(t) = X(t)\beta(t) + \epsilon(t) \quad t = 1, 2, ... \tag{1.4}$$

where $y(t)$ is the vector of measurements measured at time interval t, $X(t)$ contains the basis at interval t, $\epsilon(t)$ is the error vector and $\beta(t)$ is the vector representing the signal at interval t. And using regularized regression we can build a minimization problem for estimating the $\beta(t)$ sparsely.

$$\beta(t) = \arg_\beta \min \frac{1}{2}\|y(t) - X(t)\beta\|_2^2 + P_\lambda(|\beta|) \tag{1.5}$$

Noticing when t is fixed, the problem(1.5) reduces to (1.3). In[2, 13, 37, 38], people suggest using weighted lasso penalty in (1.3), which is formulated as:

$$P_\lambda(|\beta|) = W\|\beta\|_1 \tag{1.6}$$

where W is a diagonal matrix with positive weights. In traditional lasso penalty, all diagonal elements are the same. Some people suggest to solve a weighted lasso to further enhance the sparsity[12]. Substitute (1.6) into (1.5) and we get the

following weighted l1-norm minimization problem:

$$\beta(t) = \arg_\beta \min \frac{1}{2}\|y(t) - X(t)\beta\|_2^2 + W\|\beta\|_1 \tag{1.7}$$

In (1.7), the l2 term(least square part) keeps the solution close to the measurements and the l1 term(weighted lasso part) enforces the solution to have a sparse structure in the solution[48]. The optimization problem (1.7) is convex and can be solved by a lot of gradient methods[4, 5, 9, 43, 44] and homotopy or path methods[15, 33, 35, 41]. However, there are several deficiencies in the literatures: (1) lasso will reach a biased estimator[16]; (2) design efficient gradient methods for streaming data is very sophistic;(3) Homopoty or path methods don't have a polynomial complexity bound[29].

In this paper we propose a novel framework to recovery time varying signal: (1) we use fold concave penalty, such as, SCAD penalty which has a better statistical properties; (2) an approximated path method are designed, which is a polynomial algorithm; (3) instead of updating one measurement at one time, our method can also update a batch of measurements at once. The following sections in this chapter, we first do a review on ordinary regression and regularized regression. Then we show some popular methods to solve regularizized regression problem as well as regularizized regression with online updating situation.

1.2 Ordinary least squares regression and Kalman Filter

The ordinary least squares regression(OLS) is a very popular regression or linear inverse method. The goal of OLS is to find a minimizer of the residual sum of squared error(RSS).

$$\min \sum_{i=1}^{m}(y_i - \sum_{j}^{n} x_{ij}\beta_j)^2 \tag{1.8}$$

Where $\beta_j,\ j = 1, 2, ..., n$ are the parameters we want to estimate, $y_i,\ i = 1, 2, ..., m$ are the responses, $[x_{i1}, x_{i2}, ..., x_{in}],\ i = 1, 2, ..., m$ are the measurements. In matrix notation, we can get a simpler form:

$$\min\ \|X\beta - y\|_2^2 \tag{1.9}$$

The solution to (1.9) is:

$$\beta = (X^T X)^{-1} X^T y \tag{1.10}$$

OLS requires the signal β being constant or stable enough, while the Kalman Filter addresses a more general problem of estimating a vary signal β_t of a discrete-time or continue-time process. It assumes the signal β following a linear Gauss-Markov model:

$$\beta_k = A_k \beta_{k-1} + v_{k-1} \tag{1.11}$$

with some measurements at time t:

$$y_k = X_k \beta_k + \epsilon_k \tag{1.12}$$

Where $v_k,\ k = 1, 2, ...$ and $\epsilon_k,\ k = 0, 1, ...$ are iid Gaussian random variables, which v_k follows $N(0, Q)$ and ϵ_k follows $N(0, P)$.
The process of Kalman Filter is similar to feedback control: we first estimate the signal β_k at time k and then get feedback from minimizing measurements errors. The first stage is called time update, which is responsible for getting a prediction of current signal and the error covariance estimates for the next part. The second stage is called the measurement update which is using the new measurements to improve the estimates of current signal as well as the error covariance. The algorithm is constructed as a type of "predictor-corrector" procedure as shown below:

Algorithm 1 The Discrete Kalman Filter Algorithm

$k = 1$
while $k < K_{\max}$ **do**
  %the time update
  $\hat{\beta}_k^- = A_k \hat{\beta}_{k-1}$
  $P_k^- = A_k P_{k-1} A_k^T + Q$
  %the measurement update
  $K = P_k^- X_k^T (X_k P_k^- X_k^T + P)^{-1}$
  $\hat{\beta}_k = \hat{\beta}_k^- + K(y_k - X_k \hat{\beta}_k^-)$
  $P_k = (I - K X_k) P_k^-$
  $k = k + 1$
end while

Where $\hat{\beta}_k^-$ is the prior estimate at time k given knowledge of the process prior to time k and $\hat{\beta}_k$ is the posteriori state estimate at step k given measurements

$X_k,\ y_k$. The priori and posterior estimate errors are $e_k^- = x_k^- - \hat{x}_k^-$ and $e_k = x_k - \hat{x}_k$. $P_k^- = E[e_k^- e_k^{T-}]$ is the prior estimate error covariance and $P_k = E[e_k e_k^T]$ is the posterior estimate error covariance. More details can be found in [10].

1.3 Regularization

The idea of regularization was first introduced by Tikhonov[6], which aimed at solving ill-condition matrix inverse problem. Later, Hoerl and Kennard developed ridge regularization[22]. The formulation of ridge regression is:

$$\min \frac{1}{2}\|X\beta - y\|_2^2 + \lambda\|\beta\|_2^2 \tag{1.13}$$

The estimator of ridge least square regression is:

$$\hat{\beta}^* = (X^T X + \lambda I)^{-1} X^T y \tag{1.14}$$

Notice that when $\lambda > 0$, $X^T X + \lambda I$ is always invertible and if λ is small enough, the mean error of solution is very close to ordinary least square(OLS) regression but the prediction mean error could be much smaller than OLS. The ridge regularization won't reach a sparse solution. To enhance shrinkage, Tibshirani proposed lasso regularization[40], which use l1 norm to replace the l2 norm in the ridge regression:

$$\min \frac{1}{2}\|X\beta - y\|_2^2 + \lambda\|\beta\|_1 \tag{1.15}$$

The lasso regression can lead to a sparse and unique solution. It is also a convex problem and there are many methods to solve lasso problem. But lasso has 2 defects: (1) the number of significant parameters can't be more than the sample size; (2) it is a biased estimator. To solve the first defect, people proposed the Elastic Net regularation[49]. The formulation of Elastic Net regression is:

$$\min \frac{1}{2}\|X\beta - y\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\beta\|_2^2 \tag{1.16}$$

The solution of Elastic Net is not sparse as lasso solution but it can yield the solution that the total number of significant parameters is more than the sample size.

To solve the second defect, Fan and Li invented SCAD penalty[16], Zhang proposed MCP penalty[45], Candes, Wakin and Boyd suggested log penalty[12] and Mohimani et al used exponential penalty[31], etc. The idea of these penalties is to connect the lasso with best subset selection. The formulation of best subset selection is:

$$\min \frac{1}{2}\|X\beta - y\|_2^2 + \lambda\|\beta\|_0 \tag{1.17}$$

where

$$\|\beta_i\|_0 = \begin{cases} 1 & \alpha \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{1.18}$$

Where $i = 1, 2, ..., n$. $\|\beta\|_0$ just equals the number of its non-zero coefficients. The best subset is unbiased and sparse. Unfortunately, the known methods that can exactly solve (1.17) are combinatorial, namely NP-hard problem[14]. SCAD is a spline function that begins with lasso at around 0 and when parameter is large enough, it turns to be the best subset selection. The SCAD penalty function is defined as:

$$P_\lambda^{'}(\beta) = \lambda I_{\beta \leq \lambda} + \frac{(a\lambda - \beta)_+}{a - 1} I_{\beta > \lambda} \tag{1.19}$$

where $a > 2$ and in[16], the author suggested $a = 3.7$.
Similarly, if the spline function begins with lasso only at $\beta = 0$ and then goes towards to the best subset selection, it becomes MCP. The definition of the MCP is

$$P_\lambda^{'}(\beta) = \left(\lambda - \frac{t}{a}\right)_+ \tag{1.20}$$

Unlike the lasso, these two penalty functions do not require the irrepresentable condition[30, 46, 48] to reach the variable selection and correct the system bias of lasso method[16, 45].
Some other people suggest using log penalty function[12]:

$$P_\lambda(\beta) = \lambda \log(|\beta| + \epsilon) \tag{1.21}$$

Where ϵ is a small positive number, which is used to keep the (1.21) well defined when β is around zero. According to the approximation equation:

$$\|\beta_i\|_0 \approx \frac{\|\beta_i\|_1}{|\beta_i|} \tag{1.22}$$

Via the first order approximation, we can connect the log penalty with the best subset selection:

$$P_\lambda(\beta) = \lambda \log(|\beta| + \epsilon) \approx \lambda \frac{|\beta|}{|\hat{\beta} + \epsilon|} \approx \lambda \frac{\|\beta\|_1}{|\hat{\beta}|} \approx \lambda \|\beta\|_0 \tag{1.23}$$

The motivation of exponential penalty is very similar to log penalty function. The definition of exponential penalty function is:[31]

$$P_\lambda(\beta) = 1 - \exp(-\frac{1}{2}\lambda\beta^2) \tag{1.24}$$

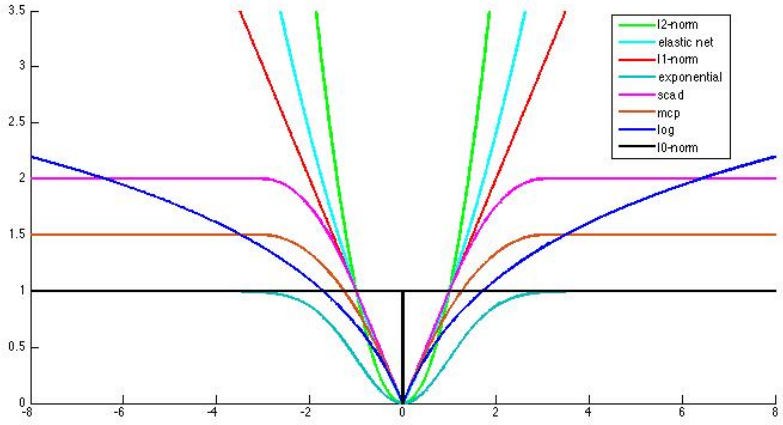

Figure 1.1: Plot of penalty function value

From the figure 1.1, we can find when β is around zero, l1-norm, SCAD, MCP and log penalty are closer to l0-norm penalty and when β is large enough, log penalty becomes further away from l0-norm. Therefore, according to the function value, SCAD and MCP may perform more like l0-norm. And from the figure 3.2, SCAD and MCP are also two of best approximated l0-norm both when β is around zero and large enough. In this paper, we prefer SCAD because when β is around zero, SCAD works more like l1-norm penalty, which has more power to force parameters becoming zero. It may make solution more stable numerically.

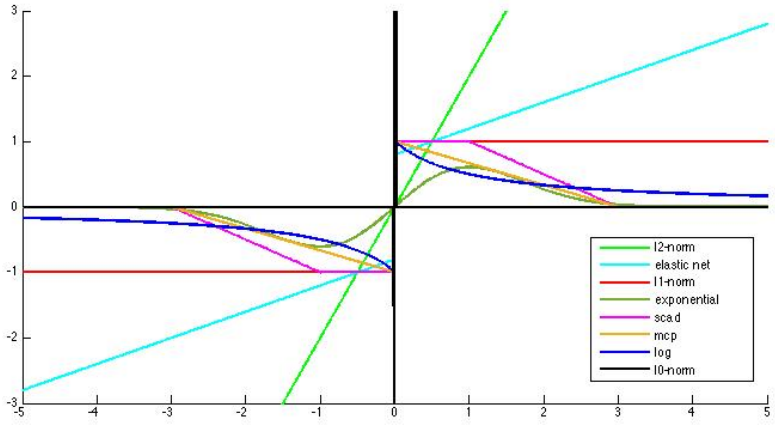

Figure 1.2: Plot of the derivative of penalty function

1.4 Algorithm for regularization regression

In general, regularation regression is nonconvex optimization problem. There are multiple local optimal solutions. The computation methods is much more involved. Several algorithms have been proposed for computing SCAD and MCP penalty problems. Fan and Li[16] set up the local quadratic approximation algorithm, which turns the nonconvex problem into solving weighted ridge regression iterately. Zou and Li[50] proposed a local linear approximation(LLA) algorithm which focus on using a sequence of weighted lasso problem to approximate the solution of nonconvex problem. Both local quadratic approximation and local linear approximation fall in the scheme of MM algorithm[25, 26]. Liu et al[24] used integer programming technology built an algorithms to solve the global optimal solution. Zhang[45] proposed a PLUS algorithm for MCP and proved the oracle property. The oracle solution is:

$$\hat{\beta}^{\text{oracle}} = (\hat{\beta}^{\text{oracle}}_{\Gamma}, 0) = \arg_{\beta} \min_{\beta_{\Gamma^c}=0} \text{loss}(\beta) \tag{1.25}$$

Where Γ contains all the index of true significant parameters and loss(.) is the loss function. For least square loss function, the oracle solution is unique and:

$$\nabla_{\Gamma} \text{loss}(\hat{\beta}^{\text{oracle}}) = 0. \tag{1.26}$$

Recently, Fan, Xue and et al shows that the local solution can be solved from LLA

Algorithm 2 The Local Linear Approximation(LLA) Algorithm

Require: $i = 0,\ \hat{\beta}_{(i)} = \hat{\beta}^{\text{initial}},\ \hat{\beta}_{(i+1)} = \hat{\beta}_{(i)} + \epsilon$
  while $\|\hat{\beta}_{(i+1)} - \hat{\beta}_{(i)}\| \geq \epsilon$ **do**
    $w_{(i)} = \{w_{1,(i)}, ..., w_{n,(i)}\} = \{P_\lambda^{'}(\beta_{1,(i)}), P_\lambda^{'}(\beta_{2,(i)})..., P_\lambda^{'}(\beta_{n,(i)})\}$
    $\hat{\beta}_{(i+1)} = \arg_\beta \min \text{loss}(\beta) + w_{(i)}\|\beta\|_1$
    $i = i + 1$
  end while

algorithm and with a probability this solution would be the oracle solution. The lower bound of this probability is $1 - \delta_0 - \delta_1 - \delta_2$[17]

$$\delta_0 = Pr(\|\hat{\beta}^{\text{initial}} - \beta^*\|_{\max} > a_0\lambda) \tag{1.27}$$

$$\delta_1 = Pr(\|\nabla_{\Gamma^c}\text{loss}(\hat{\beta}^{\text{oracle}})\|_{\max} > a_1\lambda) \tag{1.28}$$

$$\delta_2 = Pr(\|\hat{\beta}_\Gamma^{\text{oracle}} - \beta_\Gamma^*\|) \leq a\lambda \tag{1.29}$$

where a_0, a_1, a_2 are some constants related to the regularation function $P_\lambda(.)$:

$$P_\lambda^{'}(0) = P_\lambda^{'}(0_+) \geq a_1\lambda \tag{1.30}$$

$$P_\lambda^{'}(\beta) \geq a_1\lambda,\ \text{for}\ \beta \in (0, a_2\lambda] \tag{1.31}$$

$$P_\lambda^{'}(\beta) = 0,\ \text{for}\beta \in [a\lambda, \infty); a > a_2 \tag{1.32}$$

The subproblem in algorithm 2 falls in l1-minimization problem. It can be solved using various of methods listed as flows:

1.4.1 Interior Point Method for L1-minimization Problem

The first method is interior point method. To implement it, we need to replace the l1-norm with inequality constraints[27]:

$$\min \frac{1}{2}\|X\beta - y\|_2^2 + \lambda\sum_1^n u_i$$

$$\text{s.t.}\ -u_i \leq x_i \leq u_i,\ i = 1, 2, ...n$$

building a log-barrier:

$$\Phi(x,u) = -\sum_{i=1}^{n}\log(x_i+u_i) - \sum_{i=1}^{n}\log(u_i-x_i) \tag{1.33}$$

Then the central path contains a unique minimizer $(x^*(t), u^*(t))$ for the convex function

$$\phi(t,x,u) = \|X\beta - y\|_2^2 + \lambda\sum_{i=1}^{n}u_i + \frac{1}{t}\Phi(x,u) \tag{1.34}$$

as t from 0 to $+\infty$. Using Newton's method or first order methods to solve the subproblem, we will reach the final solution in polynomial steps.

1.4.2 Iterative Shrinkage-Thresholding Algorithm for L1-minimization Problem

Iterative Shrinkage-Thresholding Algorithm(ISTA) is a kind of proximal gradient methods. Given a starting point β_k, we first *prox* β_k on the l2 term:

$$\gamma_k = \beta_k - 2tX^T(X\beta_k - b) \tag{1.35}$$

Where t is a appropriate step size. Then *prox* y_k on the l1 term:

$$\beta_{k+1} = (|\gamma_k| - \lambda)_+\text{sign}(\gamma_k) \tag{1.36}$$

ISTA converges at a rate $O(\frac{1}{\epsilon})$. Using Nestrov's technique, we can enhance the convergence rate to $O(\frac{1}{\sqrt{\epsilon}})$[5].

1.4.3 Coordinate decent method for L1-minimization Problem

Coordinate decent(CD) is another very popular method for solving l1-minimiztion, in which we *prox* on each coordinate of β at a time instead of all at once.

$$\gamma_k = \frac{X_i^T(y - A_{i^c}\beta_{i^c})}{A_i^T A_i}$$
$$\beta_k = (|\gamma_k| - \lambda/\|A_i\|_2^2)_+\text{sign}(\gamma_k)$$

Repeating this for $k = 1, 2, ...n, 1, 2, ...$ until the result converges. Although the convergence rate is $O(\frac{1}{\epsilon})$[36], in the test it has a much better performance than FISFA and interior point method[18].

1.4.4 Alternating Direction Method of Multipliers for L1-minimization Problem

To use Alternating Direction Method of Multipliers(ADMM) procedure, we need to write the l1-minimization problem as:

$$\begin{aligned} &\min \frac{1}{2}\|\|X\beta - y\|_2^2 + \lambda\|\gamma\|_1 \\ &\text{s.t. } \beta - \gamma = 0 \end{aligned}$$

The ADMM becomes:

$$\beta_{k+1} = (X^T X + \rho I)^{-1}(A^T b + \rho(\gamma_k - u_k)) \tag{1.37}$$

$$\gamma_{k+1} = (\beta_{k+1} + u_k - \lambda/\rho)_+ \text{sign}(\beta_{k+1} + u_k) \tag{1.38}$$

$$u_{k+1} = u_k + \beta_{k+1} - \gamma_{k+1} \tag{1.39}$$

Note that $X^T X + \rho I$ is always invertible, since $\rho > 0$. The β-update is just a ridge regression (l2-norm regularized least squares regression) computation, so ADMM can be interpreted as a method for solving the l1-minimization problem by iteratively solving the ridge regression[9].

1.4.5 Homotopy Method for L1-minimization Problem

The homotopy method is a kind of active set methods. We start with the subgradient optimal condition:

$$0 \in X^T(X\beta - y) - \lambda\partial\|\beta\|_1 \tag{1.40}$$

where

$$\partial\|\beta\|_1 \begin{cases} = \text{sign}(\beta) & \beta \neq 0 \\ -1 \leq \partial\beta \leq 1 & \beta = 0 \end{cases} \tag{1.41}$$

Supposing we now have a solution pair (λ^t, β^t) of (1.40), the active set Γ is defined as:

$$\Gamma = \{i|\beta_i \neq 0, i = 1, 2, ..., n\} \tag{1.42}$$

Separating (1.40) according to Γ:

$$X_\Gamma^T(X_\Gamma\beta_\Gamma - y) - \lambda\partial\|\beta_\Gamma\|_1 = 0 \tag{1.43}$$

$$X_{\Gamma^c}^T(X_\Gamma\beta_\Gamma - y) - \lambda\partial\|\beta_{\Gamma^c}\|_1 = 0 \tag{1.44}$$

After some transformation:

$$\beta_\Gamma = (X_\Gamma^T X_\Gamma)^{-1}(\lambda\partial\|\beta_\Gamma\|_1 + X_\Gamma^T y) \tag{1.45}$$

$$\partial\|\beta_{\Gamma^c}\|_1 = \frac{1}{\lambda}X_{\Gamma^c}^T(X_\Gamma(X_\Gamma^T X_\Gamma)^{-1}(\lambda\partial\|\beta_\Gamma\|_1 + X_\Gamma^T y) - y) \tag{1.46}$$

We can calculate out three sets:

$$\Lambda_0 = \{\lambda_i|\beta_i = 0, i \in \Gamma\} \tag{1.47}$$

$$\Lambda_+ = \{\lambda_i|\beta_i = 1, i \in \Gamma^c\} \tag{1.48}$$

$$\Lambda_- = \{\lambda_i|\beta_i = -1, i \in \Gamma^c\} \tag{1.49}$$

$$\tag{1.50}$$

Setting $\lambda_{t+1} = \min\{\Lambda_0, \Lambda_+, \Lambda_-\}$, by checking at λ_{t+1} and Γ, (1.40) will still hold. And at $\lambda = \lambda_{t+1}$, we will either have $\beta_i = 0, i \in \Gamma$ or $\partial\|\beta_i\|_1 = \pm 1, i \in \Gamma^c$. Then we can update the active set Γ as well as $\partial\|\beta_\Gamma\|_1$ with (1.40) holding and $|\lambda_{t+1} - \lambda| < |\lambda_t - \lambda|$. So along the path of $\{\lambda_1, \lambda_2, ..., \lambda_i, ...\}$, we will finally reach the optimal solution.

We may start with $\lambda_t X^T y$ and $\Gamma = \phi$ or $\lambda_t = 0$ and $\Gamma = 1, 2, ..., p$ when sample size n is large than the parameter numbers p.

This method doesn't have a polynomial complexity bound[29] but when the amount of significant parameters is small, they can still be very efficient.

1.5 Online Updating for L1-minimization Problem

For the methods mentioned in 1.4.1-1.4.4, the online updating situation has no special designs compared to traditional methods. We just use the old solution from the previous time as a warm start, update the formulation of the old problem and run the algorithm for the updated problem with the warm start until it converges. We decide not to further discuss and only focus on homotopy type of design.

1.5.1 Homotopy path along the new observation

Many people have discussed using homotopy method to handle the situation with adding one observation at a time, [2, 19] for example. The model they based on is:

$$\min \frac{1}{2}\|X\beta - y\|_2^2 + \lambda\|\beta\|_1 + \frac{t}{2}\|x_{new}\beta - y_{new}\|_2^2 \tag{1.51}$$

When $t = 0$, (1.51) is just the old problem and if we increase t to 1, it means we add a new observation to the old problem. So the goal is to find a path for t from 0 to 1. The optimal condition for (1.51) is:

$$0 \in X^T(X\beta - y) + tx_{new}^T(x_{new}\beta - y_{new}) + \lambda\partial\|\beta\|_1 \tag{1.52}$$

where $\partial\|\beta\|_1$ follows (1.41). Assuming at $t = t_i$, we have $\Gamma = \{i|\beta_i \neq 0, i = 1, 2, ..., p\}$. We can separate (1.52) into two parts:

$$X_\Gamma^T(X_\Gamma\beta_\Gamma - y) + tx_{new,\Gamma}^T(x_{new,\Gamma}\beta_\Gamma - y_{new}) + \lambda\partial\|\beta_\Gamma\|_1 = 0 \tag{1.53}$$

$$X_{\Gamma^c}^T(X_\Gamma\beta_\Gamma - y) + tx_{new,\Gamma^c}^T(x_{new,\Gamma}\beta_\Gamma - y_{new}) + \lambda\partial\|\beta_{\Gamma^c}\|_1 = 0 \tag{1.54}$$

And we will have:

$$\beta_\Gamma = (X_\Gamma^T X_\Gamma + tx_{new,\Gamma}^T x_{new,\Gamma})^{-1}(X_\Gamma^T y + tx_{new,\Gamma}^T y_{new} - \lambda\partial\|\beta_\Gamma\|_1) \tag{1.55}$$

$$\partial\|\beta_{\Gamma^c}\|_1 = \frac{1}{\lambda}\left(X_{\Gamma^c}^T(y - X_\Gamma\beta_\Gamma) + tx_{new,\Gamma^c}^T(y_{new} - x_{new,\Gamma}\beta_\Gamma)\right) \tag{1.56}$$

Using Sherman-Morrison formula, we can rewrite $(X_\Gamma^T X_\Gamma + tx_{new,\Gamma}^T x_{new,\Gamma})^{-1}$ as

$$(X_\Gamma^T X_\Gamma)^{-1} - \frac{t(X_\Gamma^T X_\Gamma)^{-1}x_{new,\Gamma}x_{new,\Gamma}^T(X_\Gamma^T X_\Gamma)^{-1}}{1 + tx_{new,\Gamma}^T(X_\Gamma^T X_\Gamma)^{-1}x_{new,\Gamma}} \tag{1.57}$$

Denote

$$\tilde{\beta_\Gamma} = (X_\Gamma^T X_\Gamma)^{-1}((X_\Gamma^T y - \lambda\partial\|\beta_\Gamma)\|_1 \tag{1.58}$$

$$\tilde{e} = x_{new,\Gamma}^T \tilde{\beta_\Gamma} - y_{new} \tag{1.59}$$

$$a = x_{new,\Gamma}^T (X_\Gamma^T X_\Gamma)^{-1} x_{new,\Gamma} \tag{1.60}$$

$$u = (X_\Gamma^T X_\Gamma)^{-1} x_{new,\Gamma} \tag{1.61}$$

Then we will have:

$$\beta_\Gamma = \tilde{\beta_\Gamma} - \frac{(t-1)\tilde{e}}{1+a(t-1)}u \tag{1.62}$$

$$\partial\|\beta_{\Gamma^c}\|_1 = -\frac{1}{\lambda}\left(X_{\Gamma^c}\tilde{e} + \frac{\tilde{e}(t-1)}{1+a(t-1)}(x_{new,\Gamma^c} - X_{\Gamma^c}^T X_\Gamma u)\right) \tag{1.63}$$

We can calculate the folloing three sets:

$$t_0 = \{t_i | \beta_i = 0, i \in \Gamma\} \tag{1.64}$$

$$t_+ = \{t_i | \beta_i = 1, i \in \Gamma^c\} \tag{1.65}$$

$$t_- = \{t_i | \beta_i = -1, i \in \Gamma^c\} \tag{1.66}$$

Setting $t_{i+1} = \min\{t_0, t_+, t_-\}$, (1.52) will still hold at t_{i+1} and Γ. And at $t = t_{i+1}$, we either have $\beta_i = 0, i \in \Gamma$ or $\partial|\beta_i| = \pm 1, i \in \Gamma^c$. Then we can update the active set Γ and $\partial\|\beta_\Gamma\|_1$, at which (1.52) holds and $t_{i+1} > t_1$. So along the path of $\{0, t_1, t_2, ..., t_i, ...1\}$, we will final reach the solution. Since we have already known the solution without new observation denoted as β_0, it is easy to check $\beta = \beta_0, t = 0$ satisfying (1.52). We can just set it as a starting point.
This method is also very easy to expand to deal with the situation of deleting an observation via varying t from 1 to 0 on an old observation.

1.5.2 Homotopy Path Along the Subgradient

Asif proposed another type homotopy algorithm for online updating situation[1]. Supposing the updated problem is:

$$\min \ f_{new}(\beta) = \frac{1}{2}\|X_{new}\beta - y_{new}\|_2^2 + \lambda_{new}\|\beta\|_1 \tag{1.67}$$

And we can get the subgradient of $f_{new}(\beta)$:

$$\partial f_{new}(\beta) = X_{new}^T(X_{new}\beta - y_{new}) + \lambda\partial\|\beta\|_1 \tag{1.68}$$

For a given initial solution β_{ini}, define u as:

$$u \in -\partial f_{new}(\beta_{ini}) = -X_{new}^T(X_{new}\beta_{ini} - y_{new}) - \lambda\partial\|\beta_{ini}\|_1 \tag{1.69}$$

The homotopy problem is formulated as:

$$\min\ g_t(\beta) = f_{new}(\beta) + (1-t)u^T\beta \tag{1.70}$$

The subgradient of $g_t(\beta)$ is:

$$\partial g(\beta) = \partial f_{new}(\beta) + (1-t)u \tag{1.71}$$

As $u \in -\partial f_{new}(\beta_{ini})$, when $t = 0$, we will have

$$0 \in u + \partial f_{new}(\beta_{ini}) = \partial g_0(\beta)$$

So at $t = 0$ we have the optimal solution β_{ini}.
Assuming at $t = t_i$, we have $\Gamma = \{i|\beta_i \neq 0,\ i = 1, 2, ...p,\ 0 \in g_{t_i}(\beta)\}$. We can separate optimal condition into two parts:

$$X_\Gamma^T(X_\Gamma\beta_\Gamma - y) + (1-t)u_\Gamma + \lambda\partial\|\beta_\Gamma\|_1 = 0$$
$$X_{\Gamma^c}^T(X_\Gamma\beta_\Gamma - y) + (1-t)u_{\Gamma^c} + \lambda\partial\|\beta_{\Gamma^c}\|_1 = 0$$

If we increase t by a small value δ, the solution moves along the direction $\partial\beta$. Where

$$\partial\beta = \begin{cases} (X_\Gamma^T X_\Gamma)^{-1}u_\Gamma & \text{on } \Gamma \\ 0 & \text{otherwise} \end{cases} \tag{1.72}$$

If we want to maintain the optimality, we will have:

$$X_\Gamma^T(X_\Gamma\beta_\Gamma - y) + (1-t)u_\Gamma + \delta(X_\Gamma^T X\partial\beta_\Gamma - u_\Gamma) + \lambda\partial\|\beta_\Gamma\|_1 = 0$$
$$X_{\Gamma^c}^T(X_\Gamma\beta_\Gamma - y) + (1-t)u_{\Gamma^c} + \delta(X_{\Gamma^c}^T X_\Gamma\partial\beta_\Gamma - u_{\Gamma^c}) + \lambda\partial\|\beta_{\Gamma^c}\|_1 = 0$$

Denoting β^* as the solution at t_i and

$$p = X_{\Gamma^c}^T(X_\Gamma \beta^* - y) + (1 - t_i)u_{\Gamma^c} \tag{1.73}$$

$$d = X_{\Gamma^c}^T X_\Gamma \partial\beta - u_{\Gamma^c} \tag{1.74}$$

We will have:

$$\delta^+ = \min\left(\frac{\partial\|\beta_i\|_1 - p_i}{d_i}, \frac{-\partial\|\beta_i\|_1 - p_i}{d_i}, \right)_+ \qquad i \in \Gamma^c \tag{1.75}$$

$$\delta^- = \min\left(\frac{-\beta_i^*}{\partial\beta_i}\right)_+ \qquad i \in \Gamma \tag{1.76}$$

The $\min(.)_+$ means the minimum positive element. δ^+ is the smallest δ that leads to a zero coefficients becoming nonzero and δ^- is the smallest δ that leads to a non-zero coefficients becoming zero.

Setting $\delta = \min\{\delta^+, \delta^-\}$, the optimal condition will hold at $t_{i+1} = t_i + \delta$ and Γ. And at $t = t_{i+1}$, we either have $\beta_i = 0, i \in \Gamma$ or $\partial|\beta_i| = \pm 1, i \in \Gamma^c$. Then we can update the active set Γ and $\partial\|\beta_\Gamma\|_1$. As $\delta > 0$, along the path of $\{0, t_1, t_2, ..., t_i, ...\}$, we will finally reach the optimal solution. Since it is easy to check $\beta = \beta_{ini}, t = 0$ satisfying optimal condition. We can just set it as a starting point.

This method has a more flexible formulation. In the updating process, we may not only add new observations but also can change any other parts, such as, the weights of penalty function.

Chapter 2 Approximate Homotopy Algorithm

The solution path of l1-minimization probelm is piecewise linear, which makes it easy to follow and compute explicitly via homotopy method. However the worst case complexity of traditional homotopy method for lasso problem is exponential[20]. To overcome it, Mairal[29] invented an approximate homotopy for ordinary lasso regression, which has a polynomial complexity bound of $O(\frac{1}{\sqrt{\epsilon}})$. Based on [29], we prove the approximate homotopy method for lasso with online updating also has a complexity bound of $O(\frac{1}{\sqrt{\epsilon}})$. The following parts in this chapter arranged as: (1) show the proof of complexity bound of approximate homotopy for ordinary lasso regression, which is used to find a good initial solution for the next stage; (2) prove the complexity bound of approximate homotopy for lasso with online updating; (3) present model and framework of approximate homotopy for least square regression with SCAD regularization.

2.1 Complexity Bound of Approximate Homotopy for Ordinary Lasso Regression

The lasso is formulated as:

$$\min f_\lambda(\beta) = \frac{1}{2}\|X\beta - y\|_2^2 + \lambda\|\beta\|_1 \tag{2.1}$$

Lemma 2.1.1 (Optimality Conditions of lasso)**.** *For (2.1), the optimal condition*

is

$$X^T(X\beta - y) + \lambda\partial|\beta| = 0 \tag{2.2}$$

The solution of (2.2) is the unique global optimal.

where

$$\partial|\beta| \begin{cases} = \text{sign}(\beta) & \beta \neq 0 \\ \in [-1, 1] & \beta = 0 \end{cases} \tag{2.3}$$

Proof. the subgradient optimality condition for (2.1) is

$$0 \in \{X^T(X\beta - y) + \lambda p, p \in \partial|\beta|\} \tag{2.4}$$

The subgradient of l1 norm is

$$\partial\|x\|_1 \begin{cases} 1 & x > 0 \\ \in [-1, 1] & x = 0 \\ -1 & x < 0 \end{cases} \tag{2.5}$$

Which is equivalent to (2.3). It indicates the the optimal solution satisfies Lemma 2.1.1.
Note that (2.1) is also a convex combination of $l1$-norm and $l2$-norm, so (2.1) is strongly convex. If we come to a solution of (2.2), it must be the unique and global optimal solution. □

With the help of Lemma 2.1.1, we can show a well-known property of lasso:

Lemma 2.1.2 (piecewise linearity of the path)**.** *For any $\lambda > 0$ and solution of (2.1), the solution path along λ is well defined, unique and continuous piecewise linear.*

Proof. The existed and uniqueness can be get from lemma 2.1.1.
Consider $\lambda_1 < \lambda_2$, which have the same support set Γ. It is easy to show that for $\lambda_1 < \lambda^{'} < \lambda_2$, the solution on $\lambda^{'}$ is

$$\beta_{\lambda^{'},\Gamma} = \theta\beta_{\lambda_1,\Gamma} + (1 - \theta)\beta_{\lambda_2,\Gamma} \tag{2.6}$$

So the solution path between λ_1 and λ_2 is a linear segment. □

The Lemma 2.1.2 makes the homotopy method very useful as we could directly move from one end of a line to another end with very few computation cost. However, in [29] the Mairal showed that there would be $(3^p - 1)/2$ line segments between the initial solution and final optimal solution in the worst case. Meiral also proposed an approximate homotopy which do not require the exact solution of (2.2). However the author only allows λ to decrease in the proof. In the following part, we will use a similar way to show a more general result that also allow λ to both increase and decrease towards the target λ .

A natural tool to guarantee the quality of approximate solution is the duality gap. The dual problem of (2.1) is:

$$\begin{aligned} \max_{\kappa} \ g_\lambda(\kappa) = -\frac{1}{2}\kappa^T\kappa - \kappa^T y \\ \text{s.t. } \|X^T\kappa\|_\infty \leq \lambda \end{aligned}$$

Given a pair of feasible primal and dual variables (β, κ), the difference $\delta_\lambda(\beta, \kappa) = f_\lambda(\beta) - g_\lambda(\kappa)$ is called the duality gap and provides a bound for the optimal gap[8]:

$$0 \leq f_\lambda(\beta) - f_\lambda(\beta^*) \leq \delta_\lambda(\beta, \kappa)$$

Where the β^* is the optimal solution of $f_\lambda(\beta)$. It shows the gap of function value between current solution and optimal solution is always smaller than the duality gap of current solution. If the duality gap is small enough, we could say the current solution is already close enough to the optimal solution. In [29], the author uses the relative duality gap criterion to guarantee the quality of solution. Here we follow the same rule:

Definition 1. ($\epsilon-$**approximate solution**)
Let $\epsilon \in [0, 1]$, a solution β is said to be an ϵ-approximate solution of (2.1) if there exists κ such that $\|X^T\kappa\|_\infty < \lambda$ and the duality gap $\delta_\lambda(\beta, \kappa) \leq \epsilon f_\lambda(x)$

Our goal is to build a path of $\epsilon-$approximate solutions and show the complexity. To reach it, we need to introduce an approximate optimality condition based on small perturbations of those given in Lemma 2.1.1.

Definition 2. ($Opt(\lambda, \epsilon)$ **condition**)

Let $\epsilon \geq 0$. A solution β satisfies the $Opt(\lambda, \epsilon)$ condition if and only if:

$$\lambda(1-\epsilon) \leq X_\Gamma^T(y - X_\Gamma\beta)sign(\beta_\Gamma) \leq \lambda(1+\epsilon)$$
$$\lambda(1-\epsilon) \leq X_{\Gamma^c}^T(y - X_{\Gamma^c}\beta) \leq \lambda(1+\epsilon)$$

Where Γ is the active set of β, which contains all the indexes of nonzero elements in β

Note that when $\epsilon = 0$, this condition reduces to the exact optimality condition of Lemma 2.1.1. We want to connect the Definitions 1 and 2. Let us consider a solution β that satisfies the $Opt(\lambda, \epsilon_1)$. Then it is easy to check that $\kappa = \frac{1}{1+\epsilon_1}(X\beta - y)$ is feasible for the dual problem and we can compute the duality gap:

$$\begin{aligned}\delta_\lambda(\beta, \kappa) &= \frac{1}{2}(1+\epsilon_1)^2\kappa^T\kappa + \lambda\|\beta\|_1 + \frac{1}{2}\kappa^T\kappa + \kappa^T y \\ &= \frac{\epsilon_1^2}{2}\kappa^T\kappa + \lambda\|\beta\|_1 + \kappa^T(y + (1+\epsilon_1)\kappa) \\ &= \frac{\epsilon_1^2}{(1+\epsilon_1)^2}\frac{1}{2}\|X\beta - y\|_2^2 + \lambda\|\beta\|_1 + \kappa^T X\beta \end{aligned} \tag{2.7}$$

Supposing the β holds $Opt(\lambda, \epsilon_1)$, we will get $\lambda\|\beta\|_1 + \kappa^T X\beta \leq 0$, then following bound would be held:

$$\delta_\lambda(\beta, \kappa) \leq \frac{\epsilon_1^2}{(1+\epsilon_1)^2} f_\lambda(\beta) \tag{2.8}$$

Theorem 2.1.3 (Complexity bound of approximated homotopy lasso)**.** *Let $\lambda_{ini} > 0$, and $\lambda \geq 0$.For all $\epsilon \in (0, 1)$, there is an ϵ-approximate path with at most $\left\lceil \frac{\log(\lambda_{ini}/\lambda)}{\sqrt{\epsilon}} \right\rceil$ steps.*

Proof. An exact solution pair $(\beta, \kappa)_\lambda$ would satisfies $Opt(\lambda(1\pm\epsilon), \frac{\epsilon}{1-\epsilon})$. Substituting it into (2.8), we will get:

$$\delta_{\lambda(1\pm\epsilon)})(\beta, \kappa) \leq \epsilon^2 \tag{2.9}$$

So for any λ' in $[\lambda(1-\sqrt{\epsilon}), \lambda(1+\sqrt{\epsilon})]$, the solution β_λ will still be ϵ-approximate solution. It means only when we vary λ' outside $[\lambda(1-\sqrt{\epsilon}), \lambda(1+\sqrt{\epsilon})]$, we will need to go on our algorithm. Otherwise, the duality gap stop criterion satisfies. So if $\lambda_{ini} > \lambda$ before we reach λ from λ_{ini}, at most the transition points we will pass are: $\{\lambda_{ini}, \lambda_{ini}(1-\sqrt{\epsilon}), \lambda_{ini}(1-\sqrt{\epsilon})^2, ..., , \lambda_{ini}(1-\sqrt{\epsilon})^k, \lambda\}$, where

$$\lambda_{ini}(1-\sqrt{\epsilon})^{k+1} \leq \lambda \tag{2.10}$$

So the number of steps is at most

$$\left[-\frac{\log(\lambda_{ini}/\lambda)}{\log(1-\sqrt{\epsilon})}\right]+1 \geq \left[\frac{\log(\lambda_{ini}/\lambda)}{\sqrt{\epsilon}}\right] \tag{2.11}$$

Similarly, if $\lambda_{ini} < \lambda$, the number of steps is at most:

$$\left[-\frac{\log(\lambda/\lambda_{ini})}{\log(1+\sqrt{\epsilon})}\right]+1 \geq \left[\frac{\log(\lambda_/\lambda_{ini})}{-\sqrt{\epsilon}}\right] = \left[\frac{\log(\lambda_{ini}/\lambda)}{\sqrt{\epsilon}}\right] \tag{2.12}$$

□

Notice that even if λ_{ini}/λ becomes a large number, the $\log(\lambda_{ini}/\lambda)$ could still be a small number and we can treat it a constant. So the complexity bound of approximate homotopy for lasso is $O(\frac{1}{\sqrt{\epsilon}})$. The procedure of approximate homotopy for lasso is shown in Algorithm.3

Algorithm 3 Approximate Homotopy Algorithm for Ordinary Lasso

Require: $i = 0$, an exact solution $\beta^i = \beta_{ini}$, its corresponding $\lambda^i = \lambda_{ini}$ and λ_{target}
  set up the active index set $\Gamma = \{j | \beta^i_j \neq 0, j = 1, 2, ..., n\}$
  while $\lambda^i \geq \lambda_{target}$ **do**
    use homotopy method mentioned in 1.4.5 to find the next λ^{i+1} towards λ_{target}, the index k and operator $\circ$
    if $\lambda^{i+1} \notin [\lambda^i(1-\sqrt{\epsilon}), \lambda^i(1+\sqrt{\epsilon})]$ **then**
      $\Gamma = \Gamma \circ k$
    else
      use one of the methods mentioned in 1.4.1-1.4.4 to find the solution β^{i+1} at $\lambda^{i+1} = \lambda(1+\sqrt{\epsilon})$ if $\lambda_{ini} < \lambda_{target}$ or at $\lambda^{i+1} = \lambda(1-\sqrt{\epsilon})$ if $\lambda_{ini} > \lambda_{target}$
      update Γ according to β^{i+1}
    end if
    $i = i + 1$
  end while

In Algorithm 3, the operator $\circ$ is used to indicate whether parameter k should enter to Γ or remove from Γ. This algorithm can be treated as a combination of traditional homotopy method and other methods. In each step, we try a homotopy iteration firstly, if it can reach a solution outside $[\lambda^i(1-\sqrt{\epsilon}), \lambda^i(1+\sqrt{\epsilon})]$, we could directly go on to the next step. And if the result of homotopy iteration remains in$[\lambda^i(1-\sqrt{\epsilon}), \lambda^i(1+\sqrt{\epsilon})]$, we will try other methods to push λ^{i+1} out of

$[\lambda^i(1-\sqrt{\epsilon}), \lambda^i(1+\sqrt{\epsilon})]$. Algorithm 3 is used as startup method, which is used to find an initial solution for the following parts.

The iteration complexity of approximate homotopy is around $O(N^2)$. For homotopy part, the main computation is inversing $X_\Gamma^T X_\Gamma$ and we could use Sherman-Morrison formula to reduce its complexity from $O(N^3)$ to $O(N^2)$. And for the other methods, the complexity may different from each other. However for the methods mentioned in 1.4.2 to 1.4.4, the iteration complexity of these methods is also on the order of $O(N^2)$. And the solution in $[\lambda^i(1-\sqrt{\epsilon}), \lambda^i(1+\sqrt{\epsilon})]$ satisfies the dual gap criterion, which indicates the current solution is very close to the optimal solution and the methods in 1.4.2-1.4.4 would converge in few steps and can be treated as a constant. So in Algorithm 3. the complexity of methods in 1.4.2-1.4.4 is still on the order of $O(N^2)$. So the total complexity for Algorithm 3 is $O(N^2/\sqrt{\epsilon})$.

2.2 Complexity Bound of Approximate Homotopy for Lasso with Online Updating

In the previous chapter, we introduce two types of homotopy algorithm for lasso with online update. Here, we will focus on the homotopy method in 1.5.2 and show this method also has $O(1/\sqrt{\epsilon})$ complexity bound. with some modifications with some modifications.
The model that we will use is:

$$\min \ f_t(\beta) = \frac{1}{2}\|X\beta - y\|_2^2 + \frac{1}{2}\|x_{new}\beta - y_{new}\|_2^2 + \lambda_{new}\|\beta\|_1 - (1-t)u\beta \quad (2.13)$$

Where

$$u = X^T(X\beta_{ini} - y) + x_{new}^T(x_{new}\beta_{ini} - y_{ini}) + \lambda_{new}\tau \quad (2.14)$$

and

$$\tau_i = \begin{cases} sign(\beta_{ini,i}) & \beta_{ini,i} \neq 0 \\ 0 & otherwise \end{cases} \quad i = 1, 2, ...p$$

Noting that the form of (2.13) is still a quadratic term plus a $l1$-norm. We could reformulate it to (2.1) plus a constant term.

Lemma 2.2.1 (equivalence of ordinary lasso)**.** *Denoting*

$$\tilde{X} = \begin{pmatrix} X \\ X_{new} \end{pmatrix},\ \tilde{y} = \begin{pmatrix} y \\ y_{new} \end{pmatrix}$$

If there is a b such that $\tilde{X}^T b = (1-t)u$, the optimal solution of (2.13) is equivalent to the optimal solution of:

$$\min\ \ f_{eq,t}(\beta) = \frac{1}{2}\|\tilde{X}\beta - (\tilde{y}+b)\|_2^2 + \lambda_{new}\|\beta\|_1 \tag{2.15}$$

Proof. expanse (2.15):

$$\begin{aligned} f_{eq,t}(\beta) =& \frac{1}{2}\|\tilde{X}\beta - (\tilde{y}+b)\|_2^2 + \lambda_{new}\|\beta\|_1 \\ =& \frac{1}{2}\|X\beta - y\|_2^2 + \frac{1}{2}\|X_{new}\beta - y_{new}\|_2^2 + (\tilde{X}\beta - \tilde{y})^T b + \frac{1}{2}\|b\|_2^2 + \lambda_{new}\|\beta\| \\ =& \frac{1}{2}\|X\beta - y\|_2^2 + \frac{1}{2}\|X_{new}\beta - y_{new}\| + \lambda_{new}\|\beta\| + \beta^T\tilde{X}^T b - \tilde{y}^T b + \frac{1}{2}\|b\|_2^2 \\ =& f_t(\beta) - \tilde{y}^T b + \frac{1}{2}\|b\|_2^2 \end{aligned}$$

So $\arg_\beta \min f_{eq,t}(\beta) = \arg_\beta \min f_t(\beta) - \tilde{y}^T b + \frac{1}{2}\|b\|_2^2 = \arg_\beta \min f_t(\beta)$ □

Now we will verify the existence of b:

Lemma 2.2.2 (Existence of b)**.** *If β_{ini} is an exact lasso solution and we choose u follow (2.14), there must be a b satisfies $\tilde{X}b = (1-t)u$*

Proof. Denoting the active set of β_{ini} as Γ. Then we can separate $\tilde{X}b = (1-t)u$ into two parts:

$$\tilde{X}_\Gamma b_\Gamma = (1-t)u_\Gamma \tag{2.16}$$

$$\tilde{X}_{\Gamma^c} b_{\Gamma^c} = (1-t)u_{\Gamma^c} \tag{2.17}$$

For an exact lasso solution, the number of significant elements must be no more than the number of sample[34]. It means $\tilde{X}_\Gamma$ is either under-determinant or full rank. So (2.16) must have a solution b_Γ^*. For (2.17), as we choose u according to (2.14), we will have $u_{\Gamma^c} = X_{\Gamma^c}^T(X_\Gamma \beta_{ini,\Gamma} - y)$. (2.17) must have a solution $b_{\Gamma^c} = X_\Gamma \beta_{ini,\Gamma} - y$.

Now we construct a

$$b = \begin{pmatrix} b_\Gamma^* \\ X_\Gamma \beta_{ini,\Gamma} - y \end{pmatrix}$$

Which satisfies $\tilde{X}b = (1-t)u$ □

After confirming the existence of equivalent problem $f_{eq,t}(\beta)$, we could begin to build the definition of Optimal condition $Opt_{eq}(t,\epsilon)$ for $f_{eq,t}(\beta)$.

Definition 3. ($Opt_{eq}(t,\epsilon)$ **condition**)
Let $\epsilon \geq 0$. A solution β satisfies the $Opt_{eq}(t,\epsilon)$ condition if and only if:

$$\lambda(1-\epsilon) \leq \tilde{X}_\Gamma^T(\tilde{y} + b - \tilde{X}_\Gamma\beta)sign(\beta_\Gamma) \leq \lambda(1+\epsilon)$$
$$\lambda(1-\epsilon) \leq \tilde{X}_{\Gamma^c}^T(\tilde{y} - \tilde{X}_{\Gamma^c}\beta) \leq \lambda(1+\epsilon)$$

Where Γ is the active set of β, which contains all the indexes of nonzero elements in β and b is a solution of $\tilde{X}^T b = (1-t)u$

With the definition of $Opt_{eq}(t,\epsilon)$, we could prove a useful lemma:

Lemma 2.2.3 (Range of approximated solution)**.** *If β is an exact solution for a given t, β satisfies $Opt_{eq}(t \pm \frac{\lambda}{C}\epsilon, \epsilon)$ where $C = \|u\|_\infty$*

Proof. As β is an exact solution, we will have:

$$\tilde{X}_\Gamma^T(\tilde{y} + b - \tilde{X}_\Gamma\beta) + \lambda sign(\beta_\Gamma) = 0$$
$$-\lambda_{\Gamma^c} \leq \tilde{X}_{\Gamma^c}^T(\tilde{y} + b - \tilde{X}_\Gamma\beta) \leq \lambda$$

So the $Opt_{eq}(t,\epsilon)$ condition could be reduced to:

$$-\lambda\epsilon \leq \tilde{X}_\Gamma^T \delta b_\Gamma \leq \lambda\epsilon \tag{2.18}$$
$$-\lambda\epsilon \leq \tilde{X}_{\Gamma^c}^T \delta b_{\Gamma^c} \leq \lambda(1+\epsilon) \tag{2.19}$$

Substituting $\tilde{X}^T b = (1-t)u$ and combine (2.18) and (2.19) up, we can get:

$$-\lambda\epsilon \leq \delta t u \leq \lambda\epsilon \tag{2.20}$$

Supposing the maximum in u is U_1 and minimum in u is L_1. To hold (2.20), we

need to have:

$$-\lambda\epsilon \leq \delta t L_1 \tag{2.21}$$

$$\delta t U_1 \leq \lambda\epsilon \tag{2.22}$$

It means

$$\delta t = \min\{\left(-\frac{\lambda}{L_1}\epsilon\right)_+, \left(\frac{\lambda}{U_1}\epsilon\right)_+\} = \frac{\lambda}{C}\epsilon$$

□

This lemma shows that within some range of an exact solution, the $Opt_{eq}(t,\epsilon)$ condition will holds without changing β. The job now we need to do is to relate $Opt_{eq}(t,\epsilon)$ condition with ϵ-approximate solution. To reach this goal, we need to first build an upper bound of duality gap for $f_{eq,t}(\beta)$. The dual problem of $f_{eq,t}(\beta)$ is:

$$\begin{aligned} &\max\ g_{eq,t}(\kappa) = -\frac{1}{2}\kappa^T\kappa - \kappa^T(\tilde{y}+b) \\ &\text{s.t. } \|\tilde{X}^T\kappa\|_\infty \leq \lambda \end{aligned} \tag{2.23}$$

Supposing now we have a solution β that satisfies $Opt_{eq}(t,\epsilon_1)$, it is easy to check $\kappa = \frac{1}{1+\epsilon_1}(\tilde{X}\kappa - (y+b))$ is feasible for the dual problem. The duality gap is:

$$\begin{aligned} \delta_{eq,t}(\beta,\kappa) &= f_{eq,t}(\beta) - g_{eq,t}(\kappa) \\ &= \frac{1}{2}(1+\epsilon_1)^2\kappa^T\kappa + \lambda_{new}\|\beta\|_1 + \frac{1}{2}\kappa^T\kappa + \kappa^T(\tilde{y}+b) \\ &= \frac{\epsilon_1^2}{(1+\epsilon_1)^2}\frac{1}{2}\|X\beta - (\tilde{y}-b)\|_2^2 + \lambda_{new}\|\beta\|_1 + \kappa^T\tilde{X}\beta \end{aligned}$$

And as β holds $Opt_{eq}(t,\epsilon_1)$, $\lambda_{new}\|\beta\|_1 + \kappa^t\tilde{X}\beta \leq 0$. We will still have an upper bound:

$$\delta_t(\beta,\kappa) \leq \left(\frac{\epsilon_1}{1+\epsilon_1}\right)^2 f_{eq,t}(\beta) \tag{2.24}$$

Theorem 2.2.4 (Complexity bound of approximated homotopy lasso with online updating)**.** *Let $t_{ini} > 0$.For all $\epsilon \in (0,1)$, there is an ϵ-approximate path with at most $\left[\frac{C(1-\sqrt{\epsilon})}{\lambda\sqrt{\epsilon}}\right]$ steps.*

Proof. From Lemma 2.2.3, we know an exact solution pair $(\beta,\kappa)_t$ would satisfies

$Opt_eq(t + \frac{\lambda_{new}}{C}\frac{\epsilon}{1-\epsilon}, \frac{\epsilon}{1-\epsilon})$. Substituting it into (2.24), we will get:

$$\delta_{eq,tt+\frac{\lambda_{new}}{C}\frac{\epsilon}{1-\epsilon}})(\beta, \kappa) \leq \epsilon^2 \tag{2.25}$$

For any $t^{'}$ in $[t, t + \frac{\lambda_{new}}{C}\frac{\sqrt{\epsilon}}{1-\sqrt{\epsilon}}]$, the solution β will still be ϵ-approximate solution. It means only when varying $t^{'}$ outside $[t, t + \frac{\lambda_{new}}{C}\frac{\sqrt{\epsilon}}{1-\sqrt{\epsilon}}]$ and we will need go on our algorithm. Otherwise, the duality gap stop criterion would be satisfied. So before we reach $t = 1$ from $t = t_{ini}$, at most the transition points we will pass are: $\{t_{ini}, t_{ini} + \frac{\lambda_{new}}{C}\frac{\sqrt{\epsilon}}{1-\sqrt{\epsilon}}, t_{ini} + 2\frac{\lambda_{new}}{C}\frac{\sqrt{\epsilon}}{1-\sqrt{\epsilon}}, ..., 1\}$, where

$$t_{ini} + k\frac{\lambda_{new}}{C}\frac{\sqrt{\epsilon}}{1-\sqrt{\epsilon}} \geq 1 \tag{2.26}$$

So the number of steps is at most

$$\left[(1 - t_{ini})\frac{C}{\lambda_{new}}\frac{1-\sqrt{\epsilon}}{\sqrt{\epsilon}}\right] \leq \left[\frac{C(1-\sqrt{\epsilon})}{\lambda_{new}\sqrt{\epsilon}}\right] \tag{2.27}$$

□

Noticing that $1 - \sqrt{\epsilon} \approx 1$. So the complexity bound of approximate homotopy for lasso is $O(\frac{1}{\sqrt{\epsilon}})$. The procedure of approximate homotopy for lasso is show in Algorithm.4

In Algorithm 4 we also combine the traditional homotopy method with other methods. First, we use Algorithm 3 to to find a startup solution, which is used to find an initial solution for the following parts. After initialization, we will receive new observations. In each of the following step, we first try a homotopy iteration, if it can reach a solution outside $[t, t + \frac{\lambda_{new}}{C}\frac{\sqrt{\epsilon}}{1-\sqrt{\epsilon}}]$, we could directly go on to the next step. And if the result of homotopy iteration remains in$[t, t + \frac{\lambda_{new}}{C}\frac{\sqrt{\epsilon}}{1-\sqrt{\epsilon}}]$, we will try other methods to push t^{i+1} out of $[t, t + \frac{\lambda_{new}}{C}\frac{\sqrt{\epsilon}}{1-\sqrt{\epsilon}}]$. The iteration complexity of approximate homotopy is around $O(N^2)$. For homotopy part, the main computation is cost by inverse $X_\Gamma^T X_\Gamma$ and we could use Sherman-Morrison formula to reduce its complexity from $O(N^3)$ to $O(N^2)$. And for the other methods, the complexity may be different from each other. However for the methods mentioned in 1.4.2 to 1.4.4, the iteration complexity of these methods is also on the order of $O(N^2)$. And the solution in $[\lambda^i(1 - \sqrt{\epsilon}), \lambda^i(1 + \sqrt{\epsilon})]$ satisfies the dual gap criterion, which

Algorithm 4 Approximate Homotopy Algorithm for lasso with online update

Require: use Algorithm 3 to find an initial solution β
  while receiving new observations **do**
    $i = 0$ a solution $\beta^i = \beta_{ini}$ from previous stage, $t = 0$
    set up the active index set $\Gamma = \{j | \beta_j^i \neq 0, j = 1, 2, ..., n\}$, calculate u
    while $t^i \leq 1$ **do**
      use the method mentioned in 1.5.2 to find the next t^{i+1} towards 1, the index k and operator $\circ$
      if $t^{i+1} \notin [t, t + \frac{\lambda_{new}}{C} \frac{\sqrt{\epsilon}}{1-\sqrt{\epsilon}}]$ **then**
        $\Gamma = \Gamma \circ k$
      else
        use one of the methods mentioned in 1.4.1-1.4.4 to find the solution β^{i+1} at $t^{i+1} = t + \frac{\lambda_{new}}{C} \frac{\sqrt{\epsilon}}{1-\sqrt{\epsilon}}$
        update Γ according to β^{i+1}
      end if
      $i = i + 1$
    end while
  end while

indicate the current solution is very close to the optimal solution and the methods in 1.4.2-1.4.4 would converge in few steps and can be treated as a constant. So in Algorithm 3. the complexity of methods in 1.4.2-1.4.4 are still on the order of $O(N^2)$. So the total complexity for Algorithm 3 is $O(N^2/\sqrt{\epsilon})$.

2.3 Framework of Approximate Homotopy for Regularized Least Square Regression with Online Updating

The regression model is :

$$\min \frac{1}{2} \| K \left(\begin{pmatrix} X \\ X_{new} \end{pmatrix} \beta - \begin{pmatrix} y \\ y_{new} \end{pmatrix} \right) \|_2^2 + \frac{1}{2} \| \beta - A\hat{\beta}_{i-1} \|_2^2 + P_\lambda(|\beta|) \tag{2.28}$$

Where K is a Kalman matrix which is used to balance the observation error $\begin{pmatrix} X \\ X_{new} \end{pmatrix} \beta - \begin{pmatrix} y \\ y_{new} \end{pmatrix}$ and the history prediction error $\beta - A\hat{\beta}_{i-1}$. The details of the algorithm is shown in Algorithm 5.

Algorithm 5 Approximate homotopy algorithm for the regression with fold concave penalty and online updating obversations

Require: use Algorithm 3 to find the initial solution β
  while receiving observations **do**
    %use Algorithm 2(LLA) to solve (2.28)
    while Algorithm 2 not converge **do**
      build local linear approximation of penalty function
      use Algorithm 4 to solve the subproblem of Algorithm 2.
    end while
    update K according to Algorithm 1
  end while

Compared to other literatures[23][19], The Algorithm 5 have several several advantages:

(1) It is a framework that solves not only lasso problem but also general penalty linear regression.

(2) It is easier to implement, in Algorithm 5, we update all the information at once, not like [23] that request to update information part by part.

(3) When active set Γ changes slowly with respect to the time, the approximate homotopy method will reach the final solution very fast. As we can prove the following Theorem:

Theorem 2.3.1. *If total number of differences between the active set of initial solution and final solution is less than* 1*, the total number of step of Algorithm 4 is less than* 2

From chapter 1.5.2, we know in each step, the homotopy algorithm just checks the next transition point in the path. If there is no difference between the active set of initial solution and final solution, we will meet no transition point between the initial solution and final solution. If there is only one difference element i, it is easy to show that in the next step i will be added to or excluded from the active set. Now the active set of current solution is the same as it of the final solution. We will reach the final solution in the next step.

For online updating situation, we could assume that the density of observations is very large such that the set of significant parameters could only change 1 at a time if there is indeed a change. Under this assumption, the Algorithm 4 will converge very fast.

(4), We use method from Kalman Filter to balance error of observations and error of history predictions. It is more robust and adaptive.

Chapter 3
Numerical test

In [29], the author shows a worst case for traditional lasso, which will have $(3^p - 1)/2$ steps if we use homotopy method. In Lemma 2.2.1, we show that online lasso problem is equivalent to the traditional lasso. Therefore we use the case in [29] to build a worst case for lasso with online updating. The numerical result for the worst case is listed as follows::

Result/Method	Approx Homopoty	Homopoty	Nesterov's method
Total Time(s)	32	132	35
Mean Error($\times 10^4$)	1.79	2.12	2.35

Table 3.1: instances $P = 20, N = 20$

We can find that for the worst case, our algorithm has a much better performance both on computation time and result accuracy.
The numerical result for a random generated case:

Use lasso			
Result/Method	Approx Homopoty	Homopoty	Nesterov's method
Total Time(s)	3.21	3.32	5.23
Mean Error($\times 10^4$)	1.99	2.35	2.25

Table 3.2: instances $P = 2000, N = 200$

Use SCAD			
Result/Method	Approx Homopoty	Homopoty	Nesterov's method
Total Time(s)4.62	5.38	5.58	6.37
Mean Error($\times 10^4$)	1.23	1.53	1.63

Table 3.3: instances $P = 2000, N = 200$

Although time consuming doesn't improve a lot, our method still reaches a smaller mean error compared to other methods for both lasso and SCAD penalty.

3.1 Urban travel time estimation

We can also apply the algorithm to arterial traffic estimation on a simulated urban traffic network with 1000 links. We generate 200 vehicles that are traveling on the links. They will report their travel trajectories within some time duration. Each trajectory is converted in a vector $x_i \in [0, 1]^m$, where m is the number of total links in the network. The jth coordinate of x_i, denoted as$a_{i,j}$, is the fraction of the link traveled by the target vehicle. We compute it as the distance traveled on the link divided by the length of the link. In particular, $x_{i,j} = 0$ means the vehicle did not travel on link j and $a_{i,j} = 1$ means that the vehicle had already passed this link.

The solution β_n represents the average travel time on each link of the network at time t_n. We use Algorithm 5 to estimate the average travel time. In this situation, we could have historical mean travel time $\hat{x}$ to calculate the Kalman matrix K.

We compare our method with several other methods[19, 23, 32]. The result shows our method Approximate homotopy(SCAD) has better in accuracy(closer to the real and oracle solution).

Result/Method	Approx Homopoty	Homopoty	Bayen's Method	Nesterov's method	Oracle solution
Total Time(s)	3.63	3.32	3.35	4.72	-
Mean Error($\times 10^4$)	2.03	2.23	2.29	2.25	1.87

Table 3.4: Computational efficiency of simulated urban traffic data

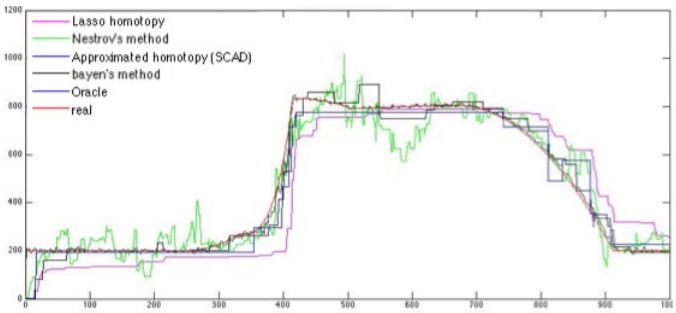

Figure 3.1: estimation of time varying signal for traffic system

In Figure 3.1, we show the how the results on a single link. We can find that the result from our method(Approximated homotopy(SCAD)) is almost closely tracking the oracle and real solution all the time. However in this picture, Bayen's method also has good solution quality. To show the difference between our methods and Bayen's method, we zoom in the picture and plot the details of a short period. We can find that our method also tracks the oracle and real solution better than Bayen's method.

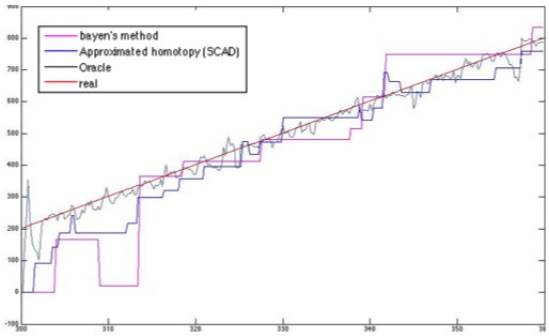

Figure 3.2: details of estimation of time varying signal for traffic system

3.2 Experiments with real images

In the aforementioned test on simulated Urban travel time, we observed that by including a linear dynamic model of a sparse, time-varying signal with the fold concave regularization provides a significantly better signal reconstruction compared to the traditional l1-regularization alone.

To evaluate the effectiveness of our method in a more realistic and streaming system, we conduct some similar of experiments on streaming signals which are generated from real-world images (shown in the top rows of Figures 3.3). In every test, all the columns of an $N \times N$ image are used to create a time-varying signal $\{x_n\}$. We assumed that adjacent columns in these images are very similar, which means $x_t \approx x_{t+1}$ $t = 1, 2, ..., N-1$. We construct the streaming, compressive measurements of $\{x_n\}$ as follows: for a given compression rate R, we generate $M = N/R$ measurements of non-overlapping x_t as $y_t = \Phi_t x_t$. We used (block-based) Daubechies 9/7 biorthogonal wavelets for sparse representation of each xt.

We estimated $\{x_n\}$ from streaming, compressive measurements in the presence and absence of the linear dynamic model (i.e., with and without the regularization term of the form $\sum_t \|x_{t+1} - x_t\|_2^2$). Results of these test are shown in the form of images in Figures 8âĂŞ11, where the color channels were compressively measured and reconstructed separately and then merged together for the purposes of display and peak signal-to-noise ratio (PSNR) computations. First rows in the figure 3.1 show the test images that were resized to $N \times N$ pixels. Second rows in Figures 3.1 show the results when each column, x_t, are independently reconstructed from its respective measurements, y_t. In other words, the recovery does not use similarities between adjacent columns. Third and fourth rows in Figures 3.1 show the results when assuming similarities between 1 and 3 adjacent columns. As we can see from the reconstructed images and their PSNR (shown in sub-captions) while independent reconstruction of each column performed poorly, a streaming framework that incorporates with linear dynamics, fold-concave and historical data (similarities in adjacent columns) provided a significantly superior recovery result.

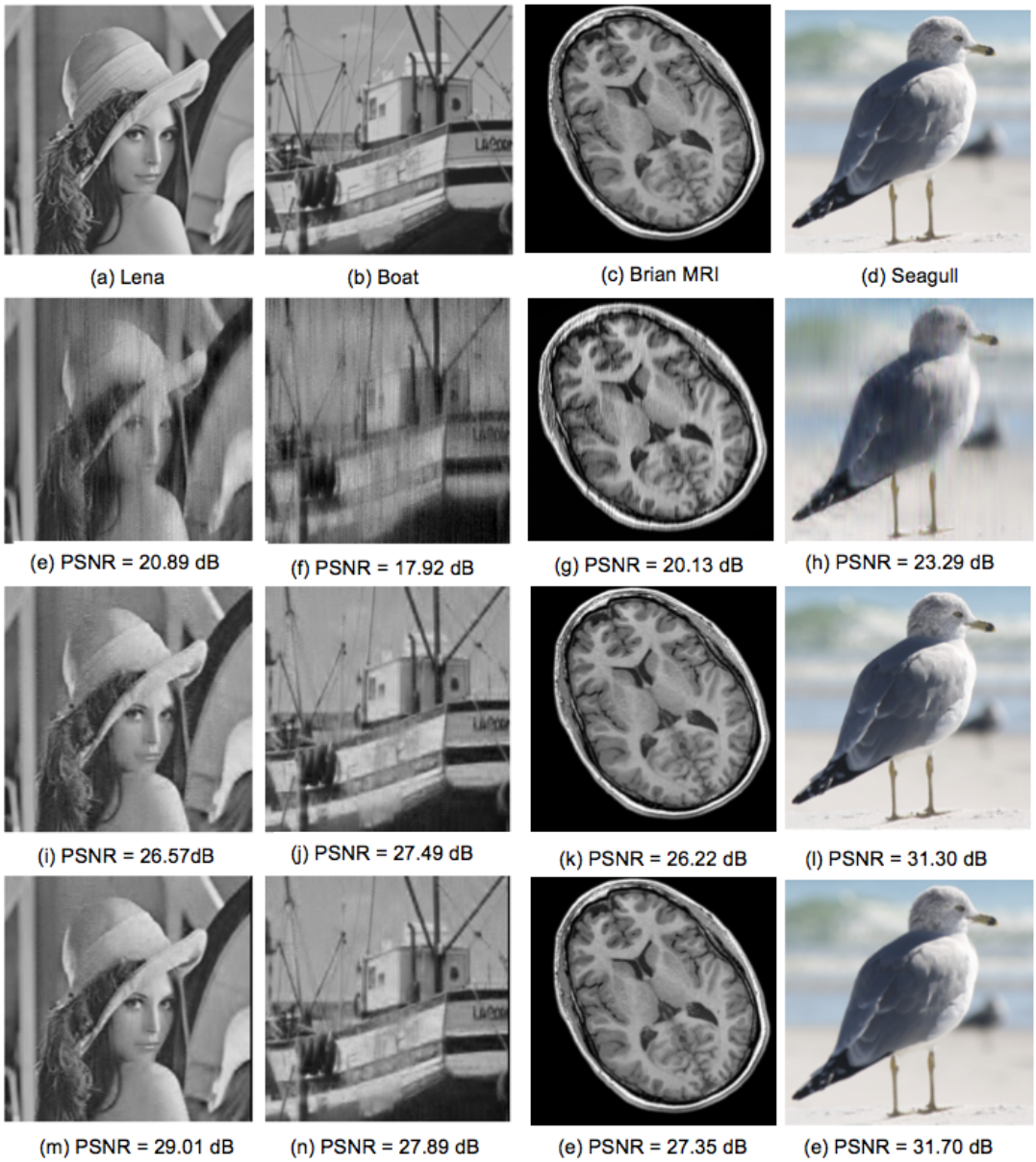

Figure 3.3: Results for the recovery of $N \times N$ images (N = 256) from column-wise random, compressive measurements with compression rate $R = 4$. (Row 1) Original images. (Row 2) Recovery of each column independent of its neighbors. (Row 3) Streaming recovery with one adjacent column. (Row 4) Streaming recovery with three adjacent columns.

Chapter 4
Conclusion

4.1 Summary of model and test results

We constructs a novel homotopy algorithm for the situation of time-varying signal with online updating. In the algorithm: (1) we use SCAD, a kind of fold concave regularation instead of l1 regularation, which has been proved to have better statistical properties; (2) the complexity bound of our method is $O(N^2/\sqrt{\epsilon})$. In addition, our method also shows a special property. No matter how dramatically the significant parameters changes, our method will converge in very few steps if the significant parameters remain significant and insignificant parameters remain insignificant. In the numerical experiments, we present our method's performance compared to some other methods in several different situations such as urban traffic travel time estimation and image recovery.

4.2 Discussion and Future Research

In the thesis, we only consider the least square loss function and there may be a chance that our method could be extended to solve problems with more general loss function. And in the Theorem 2.3.1, we only shows that when significant parameter set remains the same or with only one change at a time, our method will converge in very few steps. It is also of interest to investigate how many steps are needed when there are more than one change at a time.

Bibliography

[1] Asif, M. and Romberg, J. (2013a). Sparse recovery of streaming signals using l1 homotopy.

[2] Asif, M. S. and Romberg, J. (2013b). Fast and accurate algorithms for re-weighted-norm minimization. *Signal Processing, IEEE Transactions on*, 61(23):5905–5916.

[3] Baraniuk, R. G., Cevher, V., Duarte, M. F., and Hegde, C. (2010). Model-based compressive sensing. *Information Theory, IEEE Transactions on*, 56(4):1982–2001.

[4] Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202.

[5] Becker, S., Bobin, J., and Candès, E. J. (2011). Nesta: a fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39.

[6] Bickel, P. J., Li, B., Tsybakov, A. B., van de Geer, S. A., Yu, B., Valdés, T., Rivero, C., Fan, J., and van der Vaart, A. (2006). Regularization in statistics. *Test*, 15(2):271–344.

[7] Bishop, C. M. et al. (2006). *Pattern recognition and machine learning*, volume 4. springer New York.

[8] Borwein, J. M. and Lewis, A. S. (2010). *Convex analysis and nonlinear optimization: theory and examples*, volume 3. Springer Science & Business Media.

[9] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122.

[10] Brown, R. G., Hwang, P. Y., et al. (1992). *Introduction to random signals and applied Kalman filtering*, volume 3. Wiley New York.

[11] Candès, E. J., Romberg, J., and Tao, T. (2006). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509.

[12] Candes, E. J., Wakin, M. B., and Boyd, S. P. (2008). Enhancing sparsity by reweighted âĎŞ 1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905.

[13] Charles, A., Asif, M. S., Romberg, J., and Rozell, C. (2011). Sparsity penalties in dynamical system estimation. In *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*, pages 1–6. IEEE.

[14] Donoho, D. L. (2006). For most large underdetermined systems of linear equations the minimal ðİŞĄ1-norm solution is also the sparsest solution. *Communications on pure and applied mathematics*, 59(6):797–829.

[15] Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *The Annals of statistics*, 32(2):407–499.

[16] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360.

[17] Fan, J., Xue, L., and Zou, H. (2014). Strong oracle optimality of folded concave penalized estimation. *Annals of statistics*, 42(3):819.

[18] Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1.

[19] Garrigues, P. and Ghaoui, L. E. (2009). An homotopy algorithm for the lasso with online observations. In *Advances in neural information processing systems*, pages 489–496.

[20] Gärtner, B., Jaggi, M., and Maria, C. (2009). An exponential lower bound on the complexity of regularization paths. *arXiv preprint arXiv:0903.4817*.

[21] Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., and Tibshirani, R. (2009). *The elements of statistical learning*, volume 2. Springer.

[22] Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

[23] Hofleitner, A., El Ghaoui, L., and Bayen, A. (2011). Online least-squares estimation of time varying systems with sparse temporal evolution and application to traffic estimation. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 2595–2601. IEEE.

[24] Hongcheng liu, Tao Yao, R. L. (2014). Global solutions to folded concave penalized nonconvex learning (under review).

[25] Hunter, D. R. and Lange, K. (2004). A tutorial on mm algorithms. *The American Statistician*, 58(1):30–37.

[26] Hunter, D. R. and Li, R. (2005). Variable selection using mm algorithms. *Annals of statistics*, 33(4):1617.

[27] Kim, S.-J., Koh, K., Lustig, M., Boyd, S., and Gorinevsky, D. (2007). An interior-point method for large-scale l 1-regularized least squares. *Selected Topics in Signal Processing, IEEE Journal of*, 1(4):606–617.

[28] Li, W. and Preisig, J. C. (2007). Estimation of rapidly time-varying sparse channels. *Oceanic Engineering, IEEE Journal of*, 32(4):927–939.

[29] Mairal, J. and Yu, B. (2012). Complexity analysis of the lasso regularization path. *arXiv preprint arXiv:1205.0079*.

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

[31] Mohimani, G. H., Babaie-Zadeh, M., and Jutten, C. (2007). Fast sparse representation based on smoothed âĎŞ0 norm. In *Independent Component Analysis and Signal Separation*, pages 389–396. Springer.

[32] Nesterov, Y. et al. (2015). Complexity bounds for primal-dual methods minimizing the model of objective function. Technical report, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE).

[33] Osborne, M. R., Presnell, B., and Turlach, B. A. (2000a). A new approach to variable selection in least squares problems. *IMA journal of numerical analysis*, 20(3):389–403.

[34] Osborne, M. R., Presnell, B., and Turlach, B. A. (2000b). On the lasso and its dual. *Journal of Computational and Graphical statistics*, 9(2):319–337.

[35] Park, M. Y. and Hastie, T. (2007). L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):659–677.

[36] Saha, A. and Tewari, A. (2010). On the finite time convergence of cyclic coordinate descent methods. *arXiv preprint arXiv:1005.2146*.

[37] Salman Asif, M. and Romberg, J. (2009). Dynamic updating for sparse time varying signals. In *Information Sciences and Systems, 2009. CISS 2009. 43rd Annual Conference on*, pages 3–8. IEEE.

[38] Salman Asif, M. and Romberg, J. (2010). Dynamic updating for minimization. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):421–434.

[39] Sorenson, H. W. (1970). Least-squares estimation: from gauss to kalman. *Spectrum, IEEE*, 7(7):63–68.

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

[41] Tibshirani, R. J. (2011). *The solution path of the generalized lasso*. Stanford University.

[42] Tsaig, Y. and Donoho, D. L. (2006). Extensions of compressed sensing. *Signal processing*, 86(3):549–571.

[43] Wright, S. J., Nowak, R. D., and Figueiredo, M. A. (2009). Sparse reconstruction by separable approximation. *Signal Processing, IEEE Transactions on*, 57(7):2479–2493.

[44] Wu, T. T. and Lange, K. (2008). Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, pages 224–244.

[45] Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, pages 894–942.

[46] Zhao, P. and Yu, B. (2006). On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563.

[47] Zhu, Q., Avidan, S., and Cheng, K.-T. (2005). Learning a sparse, corner-based representation for time-varying background modelling. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 678–685. IEEE.

[48] Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429.

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

[50] Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *Annals of statistics*, 36(4):1509.