A Homotopic Method to Solve the Lasso Problems — an Improved Upper Bound of Convergence Rate

Xiaoming Huo

Georgia Institute of Technology

Joint work with Yujie Zhao

December 3, 2020

Agenda

Introduction

The Lasso Estimator

- Lasso (Tibshirani, 1996)

The Lasso Estimator

- Lasso (Tibshirani, 1996)
- Impacts...

The Lasso Estimator

- Lasso (Tibshirani, 1996)
- Impacts...
- Data generation mechanism: We have

$$y = X\beta^* + w.$$

The Lasso Estimator

- Lasso (Tibshirani, 1996)
- Impacts...
- Data generation mechanism: We have

$$y = X\beta^* + w.$$

- The Lasso estimator $\widehat{\beta}$ is

$$\widehat{\beta} = \arg\min_{\beta} \left\{ \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\}, \quad (1)$$

where parameter λ controls the trade-off between the sparsity and model's goodness of fit.

Some Discussion in solving a Lasso problem

- Essentially, an optimization problem, aiming to minimize the objective function

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

with respect to β.

Some Discussion in solving a Lasso problem

- Essentially, an optimization problem, aiming to minimize the objective function

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

  with respect to β.
- No close form for $\widehat{\beta}$.

Georgia Tech

Some Discussion in solving a Lasso problem

- Essentially, an optimization problem, aiming to minimize the objective function

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

  with respect to β.
- No close form for $\widehat{\beta}$.
- Most Lasso-algorithms are recursive with an iteration index k. That is, $\beta^{(k)}$ is the kth estimate in an iteration of a Lasso-algorithm.

Some Discussion in solving a Lasso problem

- Essentially, an optimization problem, aiming to minimize the objective function

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

  with respect to β.
- No close form for $\widehat{\beta}$.
- Most Lasso-algorithms are recursive with an iteration index k. That is, $\beta^{(k)}$ is the kth estimate in an iteration of a Lasso-algorithm.
- In this talk, we focus on *ϵ-precision*.

$$F(\beta^{(k)}) - F(\widehat{\beta}) \leq \epsilon \tag{3}$$

  and evaluate how many numerical operations are needed to achieve the above. (Note, we are not using k as the ultimate measurement of performance.)

How to compare the efficiency of Lasso-algorithms?

- We use *order of complexity* to measure the efficiency of the algorithms.

How to compare the efficiency of Lasso-algorithms?

- We use *order of complexity* to measure the efficiency of the algorithms.
- The number of numerical operations to achieve the ϵ-precision.

How to compare the efficiency of Lasso-algorithms?

- We use *order of complexity* to measure the efficiency of the algorithms.
- The number of numerical operations to achieve the ϵ-precision.
- It may be up to the sample size n, the dimension p, the precision ϵ, and other relevant factors.

How to compare the efficiency of Lasso-algorithms?

- We use *order of complexity* to measure the efficiency of the algorithms.
- The number of numerical operations to achieve the ϵ-precision.
- It may be up to the sample size n, the dimension p, the precision ϵ, and other relevant factors.
- Don't use the total number of iterations, as it may depend on what to compute within the iterations.

How to compare the efficiency of Lasso-algorithms?

- We use *order of complexity* to measure the efficiency of the algorithms.
- The number of numerical operations to achieve the ϵ-precision.
- It may be up to the sample size n, the dimension p, the precision ϵ, and other relevant factors.
- Don't use the total number of iterations, as it may depend on what to compute within the iterations.
- Don't use the running time, as it depends on the implementation and platform.

Main Result/Our Contribution

The state-of-the-art Lasso-algorithms we compare include

1. the Iterative Shrinkage-Thresholding Algorithm (ISTA)
2. the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)
3. a Coordinate Descent (CD) algorithm, and
4. the Smooth Lasso (SL)

These algorithms are representative in the literature.

Table: The available orders of complexity of four existing Lasso-algorithms and ours (in the last column) for achieving the ϵ-precision. The common factor that involves n (the sample size) and p (the dimensionality of the parameter) is omitted for simplicity.

Method	ISTA	FISTA	CD	SL	Ours
Order of complexity	$O(1/\epsilon)$	$O(1/\sqrt{\epsilon})$	$O(1/\epsilon)$	$O(1/\epsilon)$	$O\left([\log(1/\epsilon)]^2\right)$

State-of-the-Art Lasso Algorithms

Iterative Shrinkage-Thresholding Algorithms (ISTA)

- What can ISTA solve?
  ISTA aims at the minimization of a summation of two functions, $g + f$, where the first function $g : \mathbb{R}^p \to \mathbb{R}$ is continuous convex and the other function $f : \mathbb{R}^p \to \mathbb{R}$ is smooth convex with Lipschitz continuous gradient.

Iterative Shrinkage-Thresholding Algorithms (ISTA)

- What can ISTA solve?
  ISTA aims at the minimization of a summation of two functions, $g + f$, where the first function $g : \mathbb{R}^p \to \mathbb{R}$ is continuous convex and the other function $f : \mathbb{R}^p \to \mathbb{R}$ is smooth convex with Lipschitz continuous gradient.
- Lasso is a specially case of ISTA.
  If we let $g(\beta) = \lambda\|\beta\|_1$ and $f(\beta) = \frac{1}{2n}\|Y - X\beta\|_2^2$ with Lipschitz continuous gradient L taking the largest eigenvalue of matrix $X'X/n$.

Iterative Shrinkage-Thresholding Algorithms (ISTA)

- What can ISTA solve?
  ISTA aims at the minimization of a summation of two functions, $g+f$, where the first function $g:\mathbb{R}^p \to \mathbb{R}$ is continuous convex and the other function $f:\mathbb{R}^p \to \mathbb{R}$ is smooth convex with Lipschitz continuous gradient.
- Lasso is a specially case of ISTA.
  If we let $g(\beta)=\lambda\|\beta\|_1$ and $f(\beta)=\frac{1}{2n}\|Y-X\beta\|_2^2$ with Lipschitz continuous gradient L taking the largest eigenvalue of matrix $X'X/n$.
- What is the updating rule from $\beta^{(k)}$ to $\beta^{(k+1)}$, i.e., $\beta^{(k)} \to \beta^{(k+1)}$?
  It is realized by updating $\beta^{(k+1)}$ through the quadratic approximation function of $f(\beta)$ at value $\beta^{(k)}$:

$$\beta^{(k+1)} = \arg\min_{\beta} f(\beta^{(k)})+\langle(\beta-\beta^{(k)}), \nabla f(\beta^{(k)})\rangle+\frac{\sigma_{\max}(X'X/n)}{2}\|\beta-\beta^{(k)}\|_2^2+\lambda\|\beta\|_1.$$

Simple algebra shows that (ignoring constant terms in β), minimization of equation above is equal to the minimization of the following equation:

$$\beta^{(k+1)} = \arg\min_{\beta} \frac{\sigma_{\max}(X'X/n)}{2}\left\|\beta-(\beta^{(k)}-\frac{\frac{1}{n}(X'X\beta^{(k)}-X'y)}{\sigma_{\max}(X'X/n)})\right\|_2^2+\lambda\|\beta\|_1,$$

where soft-thresholding function can be used.

Iterative Shrinkage-Thresholding Algorithms (ISTA)

Algorithm 1: Iterative Shrinkage-Thresholding Algorithms (ISTA)

Input: $y_{n\times 1}, X_{n\times p}, L = \sigma_{\max}(X'X/n)$
Output: an estimator of β satisfies the ϵ-precision, noted as $\beta^{(k)}$

initialization;
$\beta^{(0)}, k = 0$
while $F(\beta^{(k)}) - F(\widehat{\beta}) > \epsilon$ **do**
    $\beta^{(k+1)} = S(\beta^{(k)} - \frac{1}{nL}(X'X\beta^{(k)} - X'y), \lambda/L)$
    $k = k + 1$
end

- Number of operations in each iteration: $O(p^2)$

Iterative Shrinkage-Thresholding Algorithms (ISTA)

Algorithm 2: Iterative Shrinkage-Thresholding Algorithms (ISTA)

Input: $y_{n\times 1}$, $X_{n\times p}$, $L = \sigma_{\max}(X'X/n)$
Output: an estimator of β satisfies the ϵ-precision, noted as $\beta^{(k)}$

initialization;
$\beta^{(0)}$, $k = 0$
while $F(\beta^{(k)}) - F(\widehat{\beta}) > \epsilon$ **do**
  $\beta^{(k+1)} = S(\beta^{(k)} - \frac{1}{nL}(X'X\beta^{(k)} - X'y), \lambda/L)$
  $k = k + 1$
end

- Number of operations in each iteration: $O(p^2)$
- Number of iterations needed to achieve the ϵ-precision: $\frac{\sigma_{\max}(X'X/n)\|\beta^{(0)} - \widehat{\beta}\|_2^2}{\epsilon}$ (Beck and Teboulle, 2009, Theorem 3.1)

Iterative Shrinkage-Thresholding Algorithms (ISTA)

Algorithm 3: Iterative Shrinkage-Thresholding Algorithms (ISTA)

Input: $y_{n\times 1}$, $X_{n\times p}$, $L = \sigma_{\max}(X'X/n)$
Output: an estimator of β satisfies the ϵ-precision, noted as $\beta^{(k)}$

initialization;
$\beta^{(0)}$, $k = 0$
while $F(\beta^{(k)}) - F(\widehat{\beta}) > \epsilon$ **do**
  $\beta^{(k+1)} = S(\beta^{(k)} - \frac{1}{nL}(X'X\beta^{(k)} - X'y), \lambda/L)$
  $k = k + 1$
end

- Number of operations in each iteration: $O(p^2)$
- Number of iterations needed to achieve the ϵ-precision: $\frac{\sigma_{\max}(X'X/n)\|\beta^{(0)}-\widehat{\beta}\|_2^2}{\epsilon}$ (Beck and Teboulle, 2009, Theorem 3.1)
- Order of complexity: $O(p^2/\epsilon)$

Fast Iterative Shrinkage-Thresholding Algorithms (FISTA)

- Motivation:
  Based on ISTA and Accelerated Gradient Descent.

Fast Iterative Shrinkage-Thresholding Algorithms (FISTA)

- Motivation:
  Based on ISTA and Accelerated Gradient Descent.
- Difference of ISTA and FISTA:
  FISTA employs the second-order Taylor expansion in equation (9) at the an auxiliary variable $\alpha^{(k)}$ to update from $\beta^{(k)}$ to $\beta^{(k+1)}$, i.e.,

$$\beta^{(k+1)} = \arg\min_{\alpha} f(\alpha^{(k)}) + \langle(\alpha - \alpha^{(k)}), \nabla f(\alpha^{(k)})\rangle + \frac{\sigma_{\max}(X'X/n)}{2}\|\alpha - \alpha^{(k)}\|_2^2 + \lambda\|\alpha\|_1,$$

where $\alpha^{(k)}$ is a specific linear combination of the previous two estimator $\beta^{(k-1)}, \beta^{(k-2)}$, i.e., we have $\alpha^{(k)} = \beta^{(k-1)} + \frac{t_{k-1}-1}{t_k}(\beta^{(k-1)} - \beta^{(k-2)})$.

Fast Iterative Shrinkage-Thresholding Algorithms (FISTA)

- Motivation:
  Based on ISTA and Accelerated Gradient Descent.
- Difference of ISTA and FISTA:
  FISTA employs the second-order Taylor expansion in equation (9) at the an auxiliary variable $\alpha^{(k)}$ to update from $\beta^{(k)}$ to $\beta^{(k+1)}$, i.e.,

$$\beta^{(k+1)} = \arg\min_{\alpha} f(\alpha^{(k)})+\langle(\alpha-\alpha^{(k)}), \nabla f(\alpha^{(k)})\rangle+\frac{\sigma_{\max}(X'X/n)}{2}\|\alpha-\alpha^{(k)}\|_2^2+\lambda\|\alpha\|_1,$$

  where $\alpha^{(k)}$ is a specific linear combination of the previous two estimator $\beta^{(k-1)}, \beta^{(k-2)}$, i.e., we have $\alpha^{(k)} = \beta^{(k-1)} + \frac{t_{k-1}-1}{t_k}(\beta^{(k-1)} - \beta^{(k-2)})$.
- FISTA falls in the framework of Accelerate Gradient Descent(AGD), because it takes additional past information to take an extra gradient step via the auxiliary sequence $\alpha^{(k)}$, which is constructed by adding a "momentum" term $\beta^{(k-1)} - \beta^{(k-2)}$ that incorporates the effect of second-order changes.

Fast Iterative Shrinkage-Thresholding Algorithms (FISTA)

Algorithm 4: Fast Iterative Shrinkage-Thresholding Algorithms (FISTA)

Input: $y_{n\times 1}, X_{n\times p}, L = \sigma_{\max}(X'X/n)$

Output: an estimator of β, noted as $\beta^{(k)}$, which satisfies the ϵ-precision.

initialization;

$\beta^{(0)}$, $t_1 = 1$, $k = 0$

while $F(\beta^{(k)}) - F(\widehat{\beta}) > \epsilon$ **do**

- $\beta^{(k)} = S(\alpha^{(k)} - \frac{1}{nL}(X'X\alpha^{(k)} - X'y), \lambda/L)$
- $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
- $\alpha^{(k+1)} = \beta^{(k)} + \frac{t_k - 1}{t_{k+1}}(\beta^{(k)} - \beta^{(k-1)})$
- $k = k + 1$

end

- Number of operations in each iteration: $O(p^2)$

Fast Iterative Shrinkage-Thresholding Algorithms (FISTA)

Algorithm 5: Fast Iterative Shrinkage-Thresholding Algorithms (FISTA)

Input: $y_{n\times 1}$, $X_{n\times p}$, $L = \sigma_{\max}(X'X/n)$

Output: an estimator of β, noted as $\beta^{(k)}$, which satisfies the ϵ-precision.

initialization;
$\beta^{(0)}$ **,** $t_1 = 1$**,** $k = 0$
while $F(\beta^{(k)}) - F(\widehat{\beta}) > \epsilon$ **do**
- $\beta^{(k)} = S(\alpha^{(k)} - \frac{1}{nL}(X'X\alpha^{(k)} - X'y), \lambda/L)$
- $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
- $\alpha^{(k+1)} = \beta^{(k)} + \frac{t_k - 1}{t_{k+1}}(\beta^{(k)} - \beta^{(k-1)})$
- $k = k + 1$

end

- Number of operations in each iteration: $O(p^2)$
- Number of iterations needed to achieve the ϵ-precision: $\frac{2\sigma_{\max}(X'X/n)\|\beta^{(0)} - \widehat{\beta}\|_2^2}{\sqrt{\epsilon}}$ (Beck and Teboulle, 2009, Theorem 4.4)

Fast Iterative Shrinkage-Thresholding Algorithms (FISTA)

Algorithm 6: Fast Iterative Shrinkage-Thresholding Algorithms (FISTA)

Input: $y_{n\times 1}$, $X_{n\times p}$, $L = \sigma_{\max}(X'X/n)$

Output: an estimator of β, noted as $\beta^{(k)}$, which satisfies the ϵ-precision.

initialization;

$\beta^{(0)}$, $t_1 = 1$, $k = 0$

while $F(\beta^{(k)}) - F(\widehat{\beta}) > \epsilon$ **do**

$\quad \beta^{(k)} = S(\alpha^{(k)} - \frac{1}{nL}(X'X\alpha^{(k)} - X'y), \lambda/L)$

$\quad t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$

$\quad \alpha^{(k+1)} = \beta^{(k)} + \frac{t_k - 1}{t_{k+1}}(\beta^{(k)} - \beta^{(k-1)})$

$\quad k = k + 1$

end

- Number of operations in each iteration: $O(p^2)$
- Number of iterations needed to achieve the ϵ-precision: $\frac{2\sigma_{\max}(X'X/n)\|\beta^{(0)} - \widehat{\beta}\|_2^2}{\sqrt{\epsilon}}$ (Beck and Teboulle, 2009, Theorem 4.4)
- Order of complexity: $O(p^2/\sqrt{\epsilon})$

Coordinate Descent (CD)

- Difference between CD and the two previous algorithms:
  The updating rule in both ISTA and FISTA is global. In contrast, Friedman et al. (2010) proposed a Lasso-algorithm which cyclically chooses one entry at a time and performs a simple analytical update through coordinate gradient descent.

Coordinate Descent (CD)

- Difference between CD and the two previous algorithms:
  The updating rule in both ISTA and FISTA is global. In contrast, Friedman et al. (2010) proposed a Lasso-algorithm which cyclically chooses one entry at a time and performs a simple analytical update through coordinate gradient descent.
- What is the updating rule from $\beta^{(k)}$ to $\beta^{(k+1)}$, i.e., $\beta^{(k)} \rightarrow \beta^{(k+1)}$?
  The updating rule from $\beta^{(k)}$ to $\beta^{(k+1)}$ in CD is that, it partially optimizes with respect to only j-th entry of $\beta^{(k+1)}$, ($j = 1 \cdots p$), where the gradient at $\beta_j^{(k)}$ in the following equation is used for the updating process.

$$\frac{\partial}{\partial \beta_j} F(\beta^{(k)}) = \frac{1}{n}\left(e_j' X' X \beta^{(k)} - y' X e_j\right) + \lambda \mathrm{sign}(\beta_\mathrm{j})$$

Coordinate Descent (CD)

Algorithm 7: Coordinate Descent(CD)

Input: $y_{n\times 1}$, $X_{n\times p}$, λ

Output: an estimator of β, noted as $\beta^{(k)}$, which satisfies the ϵ-precision.

```
initialization;
β^(0), k = 0
while F(β^(k)) − F(β̂) > ε do
    for j = 1 ··· p do
        β_j^(k+1) = S( y'Xe_j − Σ_{l≠j} (X'X)_{jl} β_k^(k), nλ ) / (X'X)_{jj}
    end
end
```

$$\beta_j^{(k+1)} = S\left(y'Xe_j - \sum_{l\neq j} (X'X)_{jl}\, \beta_k^{(k)}, n\lambda\right) / (X'X)_{jj}$$

- Number of operations in each iteration: $O(p^2)$

Coordinate Descent (CD)

Algorithm 8: Coordinate Descent(CD)

Input: $y_{n\times 1}$, $X_{n\times p}$, λ

Output: an estimator of β, noted as $\beta^{(k)}$, which satisfies the ϵ-precision.

initialization;

$\beta^{(0)}, k = 0$

while $F(\beta^{(k)}) - F(\widehat{\beta}) > \epsilon$ **do**

  for $j = 1 \cdots p$ **do**

    $\beta_j^{(k+1)} =$

    $S\left(y'Xe_j - \sum_{l\neq j} (X'X)_{jl}\, \beta_k^{(k)}, n\lambda\right) / (X'X)_{jj}$

  end

end

- Number of operations in each iteration:
  $O(p^2)$
- Number of iterations needed to achieve the ϵ-precision:
  $\frac{4\sigma_{\max}(X'X/n)(1+p)\|\beta^{(0)}-\widehat{\beta}\|_2^2}{\epsilon} - \frac{8}{p}$
  ((Beck and Tetruashvili, 2013, Corollary 3.8)

Coordinate Descent (CD)

Algorithm 9: Coordinate Descent(CD)

Input: $y_{n\times 1}$, $X_{n\times p}$, λ

Output: an estimator of β, noted as $\beta^{(k)}$, which satisfies the ϵ-precision.

initialization;

$\beta^{(0)}, k = 0$

while $F(\beta^{(k)}) - F(\widehat{\beta}) > \epsilon$ **do**

  for $j = 1 \cdots p$ **do**

    $\beta_j^{(k+1)} = S\left(y'Xe_j - \sum_{l\neq j} (X'X)_{jl}\, \beta_k^{(k)}, n\lambda\right) / (X'X)_{jj}$

  end

end

- Number of operations in each iteration: $O(p^2)$
- Number of iterations needed to achieve the ϵ-precision: $\frac{4\sigma_{\max}(X'X/n)(1+p)\|\beta^{(0)}-\widehat{\beta}\|_2^2}{\epsilon} - \frac{8}{p}$ ((Beck and Tetruashvili, 2013, Corollary 3.8)
- Order of complexity: $O(p^2/\epsilon)$

Smooth Lasso (SL)

- Main idea:
  It use a smooth function—$\phi_\alpha(u) = \frac{2}{u}\log(1 + e^{\alpha u}) - u$— to approximate ℓ_1 penalty, and Accelerated Gradient Descent (AGD) algorithm is applied after the replacement.

Smooth Lasso (SL)

Algorithm 10: Smooth Lasso (SL)

Input: $y_{n\times 1}$, $X_{n\times p}$,

$$\mu = \left[\sigma^2_{\max}(X/\sqrt{n}) + \lambda\alpha/2\right]^{-1}$$

Output: an estimator of β, noted as $\beta^{(k)}$, which satisfies the ϵ-precision.

1. **initialization;**
2. $\beta^{(0)}$ **,** $k = 0$
3. **while** $F(\beta^{(k)}) - F(\widehat{\beta}) > \epsilon$ **do**
4. $\quad w^{(k+1)} = \beta^{(k)} + \frac{k-2}{k+1}(\beta^{(k)} - \beta^{(k-1)})$
5. $\quad \beta^{(k+1)} = w^{(k+1)} - \mu\nabla F_\alpha(w^{(k)})$
6. $\quad k = k + 1$
7. **end**

- Number of operations in each iteration: $O(p^2)$

Smooth Lasso (SL)

Algorithm 11: Smooth Lasso (SL)

Input: $y_{n\times 1}$, $X_{n\times p}$,

$$\mu = \left[\sigma^2_{\max}(X/\sqrt{n}) + \lambda\alpha/2\right]^{-1}$$

Output: an estimator of β, noted as $\beta^{(k)}$, which satisfies the ϵ-precision.

```
initialization;
β^(0) , k = 0
while F(β^(k)) − F(β̂) > ε do
    w^(k+1) = β^(k) + (k−2)/(k+1) (β^(k) − β^(k−1))
    β^(k+1) = w^(k+1) − μ∇F_α(w^(k))
    k = k + 1
end
```

- Number of operations in each iteration: $O(p^2)$
- Number of iterations needed to achieve the ϵ-precision: $O(1/\epsilon)$ (Mukherjee and Seelamantula (2016))

Smooth Lasso (SL)

Algorithm 12: Smooth Lasso (SL)

Input: $y_{n\times 1}, X_{n\times p}$,

$$\mu = \left[\sigma^2_{\max}(X/\sqrt{n}) + \lambda\alpha/2\right]^{-1}$$

Output: an estimator of β, noted as $\beta^{(k)}$, which satisfies the ϵ-precision.

initialization;

$\beta^{(0)}$, $k = 0$

while $F(\beta^{(k)}) - F(\widehat{\beta}) > \epsilon$ **do**

$\quad w^{(k+1)} = \beta^{(k)} + \frac{k-2}{k+1}(\beta^{(k)} - \beta^{(k-1)})$

$\quad \beta^{(k+1)} = w^{(k+1)} - \mu \nabla F_\alpha(w^{(k)})$

$\quad k = k + 1$

end

- Number of operations in each iteration: $O(p^2)$
- Number of iterations needed to achieve the ϵ-precision: $O(1/\epsilon)$ (Mukherjee and Seelamantula (2016))
- Order of complexity: $O(p^2/\epsilon)$

Comparison between the state-of-the-art algorithms with ours

Recall the following table that has been shown earlier.

Table: The available orders of complexity of four existing Lasso-algorithms and ours (in the last column) for achieving the ϵ-precision. The common factor that involves n (the sample size) and p (the dimensionality of the parameter) is omitted for simplicity.

Method	ISTA	FISTA	CD	SL	Ours
Order of complexity	$O(1/\epsilon)$	$O(1/\sqrt{\epsilon})$	$O(1/\epsilon)$	$O(1/\epsilon)$	$O\left([\log(1/\epsilon)]^2\right)$

Our proposed Algorithm

Motivation

- We know: if the objective function is strongly convex and well conditioned, then a gradient descent method (or an accelerated gradient descent method) can achieve very fast convergence rate.

Motivation

- We know: if the objective function is strongly convex and well conditioned, then a gradient descent method (or an accelerated gradient descent method) can achieve very fast convergence rate.
- Function of the ℓ_1 norm (i.e., $||\beta||_1$) does not have derivative at the origin. (Subgradient has to be used here.)

Motivation

- We know: if the objective function is strongly convex and well conditioned, then a gradient descent method (or an accelerated gradient descent method) can achieve very fast convergence rate.
- Function of the ℓ_1 norm (i.e., $\|\beta\|_1$) does not have derivative at the origin. (Subgradient has to be used here.)
- We consider surrogate function approach:

$$F_t(\beta) = \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda f_t(\beta)$$

Motivation

- We know: if the objective function is strongly convex and well conditioned, then a gradient descent method (or an accelerated gradient descent method) can achieve very fast convergence rate.
- Function of the ℓ_1 norm (i.e., $\|\beta\|_1$) does not have derivative at the origin. (Subgradient has to be used here.)
- We consider surrogate function approach:

$$F_t(\beta) = \frac{1}{2n}\|y - X\beta\|_2^2 + \lambda f_t(\beta)$$

- Let $f_t(\beta) \to \|\beta\|_1$ as $t \to 0$.

Motivation

- We know: if the objective function is strongly convex and well conditioned, then a gradient descent method (or an accelerated gradient descent method) can achieve very fast convergence rate.
- Function of the ℓ_1 norm (i.e., $\|\beta\|_1$) does not have derivative at the origin. (Subgradient has to be used here.)
- We consider surrogate function approach:

$$F_t(\beta) = \frac{1}{2n}\|y - X\beta\|_2^2 + \lambda f_t(\beta)$$

- Let $f_t(\beta) \to \|\beta\|_1$ as $t \to 0$.
- Each $f_t(\beta)$ is strongly convex and well conditioned.

Homotopic Method

1. Recall that when $t \to 0$, we have $f_t(\beta) \to \|\beta\|_1$.

Homotopic Method

1. Recall that when $t \to 0$, we have $f_t(\beta) \to \|\beta\|_1$.
2. Design $t_0 > t_1 > t_2 > \cdots > t_k > \cdots$
   for $j = 1, 2, 3, \ldots$, (outer loop)

Homotopic Method

1. Recall that when $t \to 0$, we have $f_t(\beta) \to \|\beta\|_1$.
2. Design $t_0 > t_1 > t_2 > \cdots > t_k > \cdots$
   for $j = 1, 2, 3, \ldots$, (outer loop)
   - Minimize the following
   $$\frac{1}{2n}\|y - X\beta\|_2^2 + \lambda f_{t_j}(\beta)$$
   via a few Accelerated Gradient Descent steps (inner loop)

Homotopic Method

1. Recall that when $t \to 0$, we have $f_t(\beta) \to \|\beta\|_1$.
2. Design $t_0 > t_1 > t_2 > \cdots > t_k > \cdots$
   for $j = 1, 2, 3, \ldots,$ (outer loop)
   - Minimize the following
     $$\frac{1}{2n}\|y - X\beta\|_2^2 + \lambda f_{t_j}(\beta)$$
     via a few Accelerated Gradient Descent steps (inner loop)
   - The stopping point of the previous subproblem is the initial point of the current subproblem

Algorithm Overview

Our algorithm has two layers.

Algorithm 13: Pseudo code of the proposed Homotopy-Shrinkage algorithm

Input: A response vector y, a model matrix X, a parameter λ that relates to the Lasso

Output: an estimator of β, which satisfies the ϵ-precision

Hypter-parameter initialization initialize t

► ***Outer-Iteration:*** ◄ **while** **the precision ϵ is not achieved** **do**

    shrink t;

    ► ***Inner-Iteration:*** ◄

    use Accelerated Gradient Descent steps to minimize $F_t(\beta)$ until the precision of the inner-iteration is achieved

end

Condition for the surrogate function

To narrow down the difference between the surrogate ($F_t(\beta)$) and the origins ($F(\beta)$), $f_t(\beta)$ is designed to get more and more close to the ℓ_1 penalty when t approaches 0, i.e., $t \to 0$. The condition below lists all the requirement of $f_t(x)$.

Condition

Assume function $f_t(x)$ satisfies the following conditions.

1. When $t \to 0$, we have $f_0(x) = |x|$, where $|x|$ is the absolute value function.
2. For fixed $t > 0$, function $x \mapsto f_t(x)$ is quadratic on $[-t, t]$, here $\mapsto$ indicates that the left hand side (i.e., x) is the variable in the function in the right hand side (i.e., $f_t(x)$). We following this convention in the rest of this paper.
3. Functions $x \mapsto f_t(x)$ and $t \mapsto f_t(x)$ are C^1, i.e., continuously differentiable functions.

The Designed surrogate function

- Following the requirements in condition we mentioned, we design $f_t(x)$ in the following equation, where the input variable x is a scalar.

$$f_t(x) = \begin{cases} \frac{1}{3t^3}\left[\log(1+t)\right]^2 x^2, & \text{if } |x| \leq t, \\ \left[\frac{\log(1+t)}{t}\right]^2 |x| + \frac{1}{3|x|}\left[\log(1+t)\right]^2 - \frac{1}{t}\left[\log(1+t)\right]^2, & \text{otherwise.} \end{cases} \quad (4)$$

The Designed surrogate function

- Following the requirements in condition we mentioned, we design $f_t(x)$ in the following equation, where the input variable x is a scalar.

$$f_t(x) = \begin{cases} \frac{1}{3t^3}\left[\log(1+t)\right]^2 x^2, & \text{if } |x| \le t, \\ \left[\frac{\log(1+t)}{t}\right]^2 |x| + \frac{1}{3|x|}\left[\log(1+t)\right]^2 - \frac{1}{t}\left[\log(1+t)\right]^2, & \text{otherwise.} \end{cases} \quad (4)$$

- After the replacement of the surrogate function, we have $F_t(\beta) = \frac{1}{2n}\|y - X\beta\|_2^2 + \lambda f_t(\beta)$

The Designed surrogate function

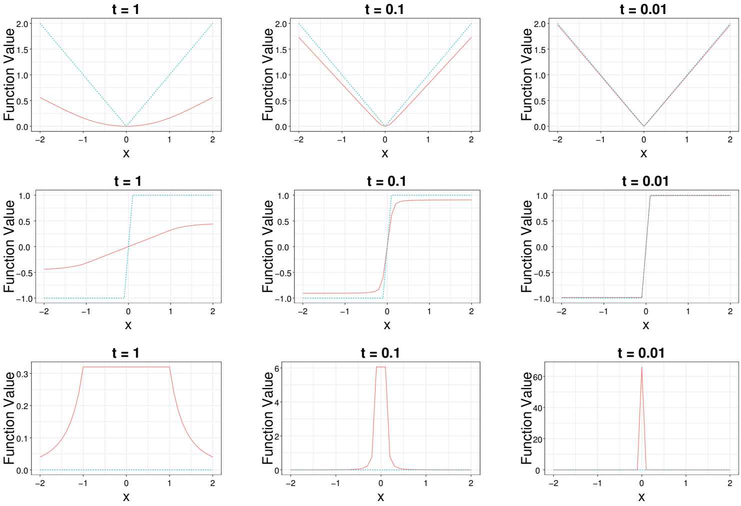

Figure: Comparison between $f_t(x)$ and $|x|$ when t changes (the first row), the comparison between their first derivative (the second row) and the comparison between their second derivative (the third row),

Detailed Algorithm

The following is the detailed algorithm.

Algorithm 2: A detailed version of our proposed algorithm

Input: $y_{n\times 1}, X_{n\times p}, \lambda, t_0, h, \epsilon, B$

Output: an estimator of β, noted as $\beta^{(k)}$, which achieves the ϵ-precision.

initialization t_0, h, $k = 1$, $\beta^{(0)} = \left[X'X + \frac{2n\lambda[\log(1+t_0)]^2}{3t_0^2} I\right]^{-1} X'y$

▶ ***Outer-Iteration:*** ◀ **while** $F(\beta^{(k-1)}) - F_{\min} > \epsilon$ **do**

- $s = 1$
- $\beta^{(k)[0]} = \beta^{(k-1)}$
- $\bar{\beta}^{(k)[0]} = \beta^{(k-1)}$
- $\widetilde{\epsilon}_k = \frac{\lambda p}{3B}[\log(1+t_k)]^2$
- ▶ ***Inner-Iteration:*** ◀ **while** $F_{t_k}(\bar{\beta}^{(k)[s-1]}) - F_{\min,k} > \widetilde{\epsilon}_k$ **do**
  - $\underline{\beta}^{(k)[s]} = (1-q_s)\bar{\beta}^{(k)[s-1]} + q_s\beta^{(k)[s-1]}$
  - $\beta^{(k)[s]} = \arg\min_\beta \left\{\gamma_s \left[\beta'\nabla F_{t_k}(\underline{\beta}^{(k)[s]}) + \mu_k V(\underline{\beta}^{(k)[s]}, \beta)\right] + V(\beta^{(k)[s-1]}, \beta)\right\}$
  - $\bar{\beta}^{(k)[s]} = (1-\alpha_s)\bar{\beta}^{(k)[s-1]} + \alpha_s\beta^{(k)[s]}$
  - $s = s + 1$
- $\beta^{(k)} = \bar{\beta}^{(k)[s]}$
- $t_k = t_{k-1}(1-h)$
- $k = k + 1$

Detailed Algorithm

Some details supporting the previous figure.

In line 2, $F_{\min} = \min_\beta F(\beta)$.
In line 4 and the rest of this paper, we use parenthesis (k) to denote the kth outer-iteration, and we use bracket $[s]$ to denote the sth inner-iteration.
In line 7, $F_{\min,k} = \min_\beta F_{t_k}(\beta)$.
In line 8, in this paper, we choose q_s as $q_s = q = \frac{\alpha_k - \mu_k/L_k}{1-\mu_k/L_k}$ for $s = 1, 2, \ldots$, where $\alpha_k = \sqrt{\frac{\mu_k}{L_k}}$. And L_k, μ_k is defined as $\|\nabla F_{t_k}(x) - \nabla F_{t_k}(y)\|_2^2 \leq L_k \|x-y\|_2$, $F_{t_k}(y) \geq F_{t_k}(x) + \nabla F_{t_k}(x)(y-x) + \frac{\mu_k}{2}\|y-x\|_2^2$.
In line 9, we choose γ_s as $\gamma_s = \gamma = \frac{\alpha}{\mu_k(1-\alpha_k)}$ for $s = 1, 2, \ldots$. Here $V(x,z)$ is defined as $V(x,z) = v(z) - [v(x) + \nabla v(x)'(z-x)]$, with $v(x) = \|x\|_2^2/2$.

Initial Point

Lemma

Suppose in a Lasso problem, we have the response vector $y \in \mathbb{R}^n$ and a model matrix $X \in \mathbb{R}^{n \times p}$. For our proposed algorithm, there exist a value t_0 that satisfies the following:

$$t_0 \in \left\{ t : \left| \sum_{j=1}^{p} M(t)_{ij} (X'y/n)_j \right| \leq t \right\}, \quad \forall i = 1, \ldots, p, \tag{5}$$

where $M(t) = \left(\frac{X'X}{n} + \frac{\lambda}{3t^3} [\log(1+t)]^2 I \right)^{-1}$. Here X' represents the transpose of matrix X, and we use this notation in the remaining of the paper. When one chooses the aforementioned t_0 as the initial point in the proposed algorithm, we have $\left|\beta_i^{(0)}\right| \leq t_0$ for any $1 \leq i \leq p$, where $\beta_i^{(0)}$ denotes the ith entry in the vector $\beta^{(0)} = M(t_0)X'y/n$.

update of the hyper-parameter

- Shrink the t_k to $t_{k+1} = t_k(1 - h)$

update of the hyper-parameter

- Shrink the t_k to $t_{k+1} = t_k(1 - h)$
- Early Stopping in the Inner-Loop: in the kth outer-iteration, the inner-iteration is stopped when

$$F_{t_k}(\beta^{(k)[s]}) - F_{\min,k} < \widetilde{\epsilon}_k,$$

where $\beta^{(k)[s]}$ denotes the iterative estimator in the sth inner-iteration of the kth outer iteration, and we have

$$F_{\min,k} = \min_{\beta} F_{t_k}(\beta).$$

Here, we set $\widetilde{\epsilon}_k = \frac{\lambda p}{3B} [\log(1 + t_k)]^2$, where B is the upper bound of $\left|\beta_i^{(k)}\right|$ for all $i = 1, 2, \ldots, p$ and $k = 1, 2, \ldots$.

Theoretical Results

Theorems

Theorem (Inner-Loop)

Recall that a Lasso problem has a response vector $y \in \mathbb{R}^n$ and a model matrix $X \in \mathbb{R}^{n \times p}$. To minimize the Lasso objective function $F(\beta) = \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda\|\beta\|_1$, we design a homotopic approach, i.e., in the kth outer-iteration of our proposed algorithm, we use AGD algorithm to minimize a surrogate function $F_{t_k}(\beta) = \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda f_{t_k}(\beta)$. Instead of converging to the minimizer of $F_{t_k}(\beta)$, we do an early stopping to control the total number of numerical operations. And we denote the early stopping estimation as $\beta^{(k)[s]}$, where s is the number of AGD-iterations (inner-iterations). We assume that for any $k = 1, 2, \ldots, s = 1, 2, \ldots$, we have $\left|\beta_i^{(k)[s]}\right| \leq B$, where $\beta_i^{(k)[s]}$ is the ith entry of vector $\beta^{(k)[s]}$ ($i = 1, 2, \ldots, p$), and B is a constant. And we further assume that our proposed algorithm stops when $t_k < \tau$. Under the above assumption, we know that in the kth outer-iteration, the condition number of function $F_{t_k}(\cdot)$ can be bounded by $\frac{3B^3 \lambda_{\max}\left(\frac{X'X}{n}\right)}{2\lambda[\log(1+\tau)]^2} + \left(\frac{B}{\tau}\right)^3$. Accordingly, after $C_1 \log(1/\widetilde{\epsilon}_k)$ inner-iterations, one is guaranteed to achieve the following precision

$$F_{t_k}(\beta^{(k)}) - F_{\min,k} \leq \widetilde{\epsilon}_k,$$

where $F_{\min,k} = \min_\beta F_{t_k}(\beta)$, $\widetilde{\epsilon}_k = \frac{\lambda p}{3B} [\log(1 + t_k)]^2$ and C_1 is a constant that does not depend on the value of t_k.

Theorems

Theorem (Number of outer-iteration)

With the conditions in Theorem 3 being satisfied, and suppose the following conditions are also satisfied:

1. *For $k = 1, 2, \ldots$, we have $t_k = t_0(1-h)^k$ where t_0, h are pre-specified.*
2. *The precision of AGD in minimizing function $F_{t_k}(\beta)$ is set as $\widetilde{\epsilon}_k = \frac{\lambda p}{3B}[\log(1+t_k)]^2$, i.e., we run the AGD until the following inequality is achieved: $F_{t_k}(\beta^{(k)[s]}) - F_{k,\min} < \widetilde{\epsilon}_k$, where quantity $\beta^{(k)[s]}$ is the iterative estimator in the sth inner-iteration at the kth outer-iteration, and recall that $F_{k,\min} = \min_\beta F_{t_k}(\beta)$.*

Then when $k \geq \frac{-1}{\log(1-h)} \log\left(\frac{\lambda p t_0(2B+1)}{\epsilon}\right)$, our proposed algorithm finds a point $\beta^{(k)}$ such that

$$F(\beta^{(k)}) - F_{\min} \leq \epsilon,$$

where $F_{\min} = \min_\beta F(\beta)$ with $F(\beta) = \frac{1}{2n}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$, which is defined in (2).

Theorems

Theorem (Main Theory)

Under the conditions that are listed in Theorem 3 and Theorem 4, we can find $\beta^{(k)}$ *such that*

$$F(\beta^{(k)}) - F_{\min} \leq \epsilon$$

with the number of numerical operations has the order of complexity

$$p^2 O\left(\left[\frac{-1}{\log(1-h)} \log\left(\frac{\lambda p t_0 (2B+1)}{\epsilon}\right)\right]^2\right).$$

Theorems

Remark

For our proposed algorithm, when $t \to 0$, we have the perdition error $\frac{1}{n}\left\|X\left(\widetilde{\beta}-\widehat{\beta}\right)\right\|_2^2 \to 0$, where $\widetilde{\beta} = \arg\min_\beta \frac{1}{2n}\|y - X\beta\|_2^2 + \lambda f_t(\beta)$ for a general t and $f_t(\beta)$ defined in (4). And $\widehat{\beta} = \arg\min_\beta \frac{1}{2n}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$.

Theorems

Remark

Suppose the model matrix X in the Lasso problem has the following three properties:

1. $\left\| \left(X_S' X_S\right)^{-1} X_S' \right\|_F$ can be bounded by a constant, where $S = \{i : \widehat{\beta}_i \neq 0, \forall i = 1, 2, \ldots, p\}$ with $\widehat{\beta} = \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$. And $\|\cdot\|_F$ is the Frobenius norm defined as $\left\|A_{m\times n}\right\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}$, where a_{ij} is the (i, j)th entry in matrix A.
2. $\left\| X_{S^C}^{\dagger} \right\|_F$ can be bounded by a constant, where S^C is the complement set of S. And $X_{S^C}^{\dagger}$ is the pseudo-inverse of matrix X_{S^C}. The mathematical meaning of pseudo-inverse is that, suppose $X_{S^C} = U\Sigma V$, which is the singular value decomposition (SVD) of X_{S^C}. Then $X_{S^C}^{\dagger} = V'\Sigma^{\dagger} U'$. For the rectangular diagonal matrix Σ, we get $\Sigma^{\dagger}$ by taking the reciprocal of each non-zero elements on the diagonal, leaving the zeros in place, and then transposing the matrix.
3. $\sigma_{\max}(\Sigma_1) < \min\{2, 2\sigma_{\min}(\Sigma_2)\}$, where $\sigma_{\max}(\Sigma_1)$ returns the maximal absolute diagonal values of matrix Σ_1, and $\sigma_{\min}(\Sigma_2)$ returns the minimal absolute diagonal values of matrix Σ_2. Matrix Σ_1 is the diagonal matrix in the SVD of matrix $\left(X_S' X_S\right)^{-1} X_S' X_{S^C} + \left(X_{S^C}^{\dagger} X_S\right)'$, i.e., $\left(X_S' X_S\right)^{-1} X_S' X_{S^C} + \left(X_{S^C}^{\dagger} X_S\right)' = U_1 \Sigma_1 V_1$. Matrix Σ_2 is the diagonal matrix of the SVD of matrix $\frac{1}{2} X_{S^C}^{\dagger} X_{S^C} + \frac{1}{2}\left(X_{S^C}^{\dagger} X_{S^C}\right)'$, i.e., $\frac{1}{2} X_{S^C}^{\dagger} X_{S^C} + \frac{1}{2}\left(X_{S^C}^{\dagger} X_{S^C}\right)' = U_2 \Sigma_2 V_2$.

Then we have $\left\|\widetilde{\beta} - \widehat{\beta}\right\|_2^2 \to 0$ when $t \to 0$.

Numerical Example

Data Generation

- We generated Gaussian data with n observations and p predictors, with each pair of predictors noted as X_j, and the explanatory matrix is noted as $X = (X_1 \cdots X_j \cdots X_p)$.

Data Generation

- We generated Gaussian data with n observations and p predictors, with each pair of predictors noted as X_j, and the explanatory matrix is noted as $X = (X_1 \cdots X_j \cdots X_p)$.
- Here we assume that X_j having the same population correlation ρ and we tried a number of ρ varying from 0 to 0.9. (We choose $\rho = 0.5$ in this section.)

Data Generation

- We generated Gaussian data with n observations and p predictors, with each pair of predictors noted as X_j, and the explanatory matrix is noted as $X = (X_1 \cdots X_j \cdots X_p)$.
- Here we assume that X_j having the same population correlation ρ and we tried a number of ρ varying from 0 to 0.9. (We choose $\rho = 0.5$ in this section.)
- Accordingly, the outcome values were generated by

$$y = X\beta + qz.$$

where the i-th ($1 \le i \le p$) entry in vector $\beta = (\beta_1 \cdots \beta_p)'$ is generated by $\beta_i = (-1)^i \exp\left(-2(i-1)/20\right)$, which are constructed to have alternating signs and to be exponentially decreasing.

Data Generation

- We generated Gaussian data with n observations and p predictors, with each pair of predictors noted as X_j, and the explanatory matrix is noted as $X = (X_1 \cdots X_j \cdots X_p)$.
- Here we assume that X_j having the same population correlation ρ and we tried a number of ρ varying from 0 to 0.9. (We choose $\rho = 0.5$ in this section.)
- Accordingly, the outcome values were generated by

$$y = X\beta + qz.$$

  where the i-th ($1 \leq i \leq p$) entry in vector $\beta = (\beta_1 \cdots \beta_p)'$ is generated by $\beta_i = (-1)^i \exp\left(-2(i-1)/20\right)$, which are constructed to have alternating signs and to be exponentially decreasing.
- Besides, $z = (z_1 \cdots z_p)'$ is the white noise with normal distribution of z_i as $N(0, 1)$ and q is chosen so that the signal-to-noise ratio is 3.0.

Simulation 1

Table: Numerical complexity of ISTA, FISTA, HS in the first simulation

	Precision ϵ								
method	0.05	0.03	0.02	0.01	0.009	0.008	0.007	0.006	0.005
	$n = 50,\ p = 20$								
ISTA	5,070	6,016	7,005	9,585	10,101	10,703	11,434	12,294	13,369
FISTA	4,781	5,117	5,453	6,461	6,685	6,797	7,021	7,133	7,357
Ours	5,478	5,478	5,479	5,479	5,479	5,479	5,479	5,479	6,005
	$n = 50,\ p = 80$								
ISTA	37,400	50,277	65,273	109,772	119,226	130,799	145,306	164,377	190,131
FISTA	31,237	34,533	37,417	45,657	47,305	48,541	50,189	52,249	55,133
Ours	30,919	30,919	32,765	34,611	34,611	34,611	36,457	36,457	36,457

[1] There is the parameters settings of our HS algorithm: $t_0 = 3$, $h = 0.1$, $\lambda = 1e-3$, $\beta^{(0)} = \mathbf{1}_{p\times 1}$.

Simulation Result

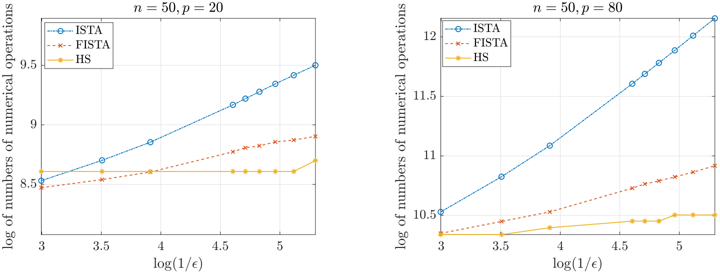

Figure: Number of Operations of ISTA, FISTA, and our algorithm under different ϵ in the first simulation

Simulation 2

Table: Numerical complexity of ISTA, FISTA, HS in the second simulation

	Precision ϵ								
method	0.05	0.03	0.02	0.01	0.009	0.008	0.007	0.006	0.005
	$n = 50,\ p = 20$								
ISTA	5,242	6,274	7,263	9,370	9,757	10,187	10,703	11,305	12,122
FISTA	4,781	5,229	5,565	6,125	6,349	6,461	6,573	6,797	7,021
Ours	5,479	5,479	5,479	6,005	6,005	6,005	6,005	6,005	6,005
	$n = 50,\ p = 80$								
ISTA	39,519	55,330	72,119	112,869	120,693	130,473	142,698	158,346	179,373
FISTA	31,649	35,769	39,065	45,657	46,893	48,129	49,365	51,013	53,485
Ours	30,918	32,763	32,763	32,763	32,763	32,763	32,763	32,763	32,763

[1] The parameters settings of our HS algorithm: $t_0 = 3$, $h = 0.1$, $\lambda = 1e - 3$, $\beta^{(0)} = \mathbf{0.1}_{p\times 1}$

Simulation 2

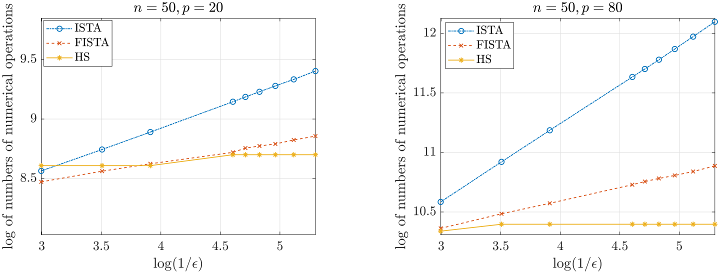

Figure: Number of Operations of ISTA, FISTA, and our algorithm under different ϵ in the second simulation.

Other Related Homotopic Ideas

- Start at some initial objective problem, which is a Lasso-type objective function:

$$\frac{1}{2n}\|y - X\beta\|_2^2 + \lambda^{(0)}\|\beta\|_1, \tag{6}$$

and then they gradually decrease the large $\lambda^{(0)}$ until the target regularization $\lambda^{(\mathrm{target})}$ is reached.

Other Related Homotopic Ideas

- Start at some initial objective problem, which is a Lasso-type objective function:

$$\frac{1}{2n}\|y - X\beta\|_2^2 + \lambda^{(0)}\|\beta\|_1, \tag{6}$$

  and then they gradually decrease the large $\lambda^{(0)}$ until the target regularization $\lambda^{(\text{target})}$ is reached.
- A close related topic: path-following approach

Other Related Homotopic Ideas

- Start at some initial objective problem, which is a Lasso-type objective function:

$$\frac{1}{2n}\|y - X\beta\|_2^2 + \lambda^{(0)}\|\beta\|_1, \tag{6}$$

  and then they gradually decrease the large $\lambda^{(0)}$ until the target regularization $\lambda^{(\text{target})}$ is reached.
- A close related topic: path-following approach
- When the path-following approaches work, it is a very nice algorithm; however, contrary to general beliefs, they only work in special cases.

Other Related Homotopic Ideas

- Start at some initial objective problem, which is a Lasso-type objective function:

$$\frac{1}{2n}\|y - X\beta\|_2^2 + \lambda^{(0)}\|\beta\|_1, \tag{6}$$

  and then they gradually decrease the large $\lambda^{(0)}$ until the target regularization $\lambda^{(\text{target})}$ is reached.
- A close related topic: path-following approach
- When the path-following approaches work, it is a very nice algorithm; however, contrary to general beliefs, they only work in special cases.
- Counterexample, where path-following conditions are violated, can be constructed.

Conclusion/The End

- A newly designed homotopic approach that can solve the Lasso-type of problem with better theoretical guarantees (on numerical complexity)

Conclusion/The End

- A newly designed homotopic approach that can solve the Lasso-type of problem with better theoretical guarantees (on numerical complexity)
- Interplay of Statistics and Computing

Conclusion/The End

- A newly designed homotopic approach that can solve the Lasso-type of problem with better theoretical guarantees (on numerical complexity)
- Interplay of Statistics and Computing
- Future work: more general results than the existing ones in the literature, taking advantage of the homotopic approaches

Conclusion/The End

- A newly designed homotopic approach that can solve the Lasso-type of problem with better theoretical guarantees (on numerical complexity)
- Interplay of Statistics and Computing
- Future work: more general results than the existing ones in the literature, taking advantage of the homotopic approaches
- **ArXiv**: https://arxiv.org/abs/2010.13934

Conclusion/The End

- A newly designed homotopic approach that can solve the Lasso-type of problem with better theoretical guarantees (on numerical complexity)
- Interplay of Statistics and Computing
- Future work: more general results than the existing ones in the literature, taking advantage of the homotopic approaches
- **ArXiv**: https://arxiv.org/abs/2010.13934
- **Email**: huo@gatech.edu

Conclusion/The End

- A newly designed homotopic approach that can solve the Lasso-type of problem with better theoretical guarantees (on numerical complexity)
- Interplay of Statistics and Computing
- Future work: more general results than the existing ones in the literature, taking advantage of the homotopic approaches
- **ArXiv**: https://arxiv.org/abs/2010.13934
- **Email**: huo@gatech.edu
- Thank You!!!