Solve Lasso

Machine Learning Course

Spring 2015

Tsinghua University

Goal

- Compare several methods for solving Lasso.

Lasso

- Problem : Lasso with no structure among covariates

$$\min_{\boldsymbol{\omega} \in \mathbb{R}^p} \|\mathbf{y} - \mathbf{X}\boldsymbol{w}\|_2^2 + \lambda \|\boldsymbol{w}\|_1, \quad \mathbf{y} \in \mathbb{R}^n, \ \mathbf{X} \in \mathbb{R}^{n \times p}$$

- Each row of **X** is a data vector.
- Each column of **X** represents a feature.
- **y** represents the output, which is a sparse construction from **X**. **w** is the construction weight.

Let $f(\boldsymbol{\omega}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{w}\|_2^2$ be the smooth part, $g(\boldsymbol{\omega}) = \lambda \|\boldsymbol{w}\|_1$ be the non-smooth part, then the former problem can be written as

$$\min_{\boldsymbol{\omega} \in \mathbb{R}^p} f(\boldsymbol{\omega}) + g(\boldsymbol{\omega})$$

Lasso

- Implement the first order methods and compare them
- Solvers
  - SubGradient Descent (SGD), **Proximal Methods (ISTA,** FISTA), **Coordinate Descent (CD)**, Quadratic Programming (QP), Cone Programming (CP), Reweighted L2 (Re-L2), Least Angle Regression (LARS), etc.

Lasso – ISTA solver

For iteration t=1 to $\cdots$

-Step1: Calculate the gradient of the smooth part

$$\nabla_{\boldsymbol{\omega}^t} f(\boldsymbol{\omega}^t) = -2\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\omega}^t)$$

-Step2: Approximate f around the current point

$$f(\boldsymbol{\omega}) \approx f(\boldsymbol{\omega}^t) + \nabla_{\boldsymbol{\omega}^t} f(\boldsymbol{\omega}^t)^\top(\boldsymbol{\omega} - \boldsymbol{\omega}^t) + \frac{L}{2}\|\boldsymbol{\omega} - \boldsymbol{\omega}^t\|_2^2$$

-Step 2.1 : Solve the problem with fixed L

$$\boldsymbol{\omega}_{\boldsymbol{L}}^* = \text{Prox}_{\frac{\lambda}{L}}(\boldsymbol{\omega}^t - \frac{1}{L}\nabla_{\boldsymbol{\omega}^t} f(\boldsymbol{\omega}^t)), \text{ where } \text{Prox}_k(x) = \text{sgn}(x)\max(|x|$$

-Step 2.2 : If L satisfies

$$f(\boldsymbol{\omega}_{\boldsymbol{L}}^*) \leq f(\boldsymbol{\omega}^t) + \nabla_{\boldsymbol{\omega}^t}^\top(\boldsymbol{\omega}_{\boldsymbol{L}}^* - \boldsymbol{\omega}^t) + \frac{L}{2}\|\boldsymbol{\omega}_{\boldsymbol{L}}^* - \boldsymbol{\omega}^t\|_2^2 \text{ , set } \boldsymbol{\omega}^{t+1} = \boldsymbol{\omega}_{\boldsymbol{L}}^*$$

and go to next iteration; else increase L by a factor and go to Ste

Lasso – CD solver

Regular version

For iteration t=1 to $\cdots$

For each covariate(feature) j, do

$$\boldsymbol{\omega}_j^{t+1} = \mathrm{Prox}_{\frac{\lambda}{2\mathbf{X}_{\cdot\mathbf{j}}^\top \mathbf{X}_{\cdot\mathbf{j}}}}(\boldsymbol{\omega}_j^t - \frac{{\mathbf{X}_{\cdot\mathbf{j}}}^\top \mathbf{X}\boldsymbol{\omega} - \mathbf{X}_{\cdot\mathbf{j}}^\top \mathbf{y}}{\mathbf{X}_{\cdot\mathbf{j}}^\top \mathbf{X}_{\cdot\mathbf{j}}})$$

Stochastic version

For iteration t=1 to $\cdots$

Randomly pick a covariate j, do

$$\boldsymbol{\omega}_j^{t+1} = \mathrm{Prox}_{\frac{\lambda}{2\mathbf{X}_{\cdot\mathbf{j}}^\top \mathbf{X}_{\cdot\mathbf{j}}}}(\boldsymbol{\omega}_j^t - \frac{{\mathbf{X}_{\cdot\mathbf{j}}}^\top \mathbf{X}\boldsymbol{\omega} - \mathbf{X}_{\cdot\mathbf{j}}^\top \mathbf{y}}{\mathbf{X}_{\cdot\mathbf{j}}^\top \mathbf{X}_{\cdot\mathbf{j}}})$$

Extension: Lasso – SGD solver

For iteration t=1 to $\cdots$

-Step1: Calculate the (sub)gradient

$$\nabla_{\boldsymbol{\omega}^t}(f(\boldsymbol{\omega}^t) + g(\boldsymbol{\omega}^t)) = -2\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\omega}^t) + \lambda \sum_i \mathrm{sgn}({\boldsymbol{\omega}^t}_i)$$

-Step2 : Decide step size α

$\alpha^t = a/(t+b),\ a, b$ are pre-defined constant.

-Step3 : Do gradient descent

$$\boldsymbol{\omega}^{t+1} = \boldsymbol{\omega}^t - \alpha^t \nabla_{\boldsymbol{\omega}^t}(f(\boldsymbol{\omega}^t) + g(\boldsymbol{\omega}^t))$$

Extension: Group Lasso

- Problem : Lasso with group structure among covariates (just solve the l1/l2 case)

$$\min_{\boldsymbol{\omega}\in\mathbb{R}^p} \|\mathbf{y}-\mathbf{X}\boldsymbol{\omega}\|_2^2 + \lambda \sum_{g\in\mathcal{G}} \|\boldsymbol{w_g}\|_2, \quad \mathbf{y}\in\mathbb{R}^n,\ \mathbf{X}\in\mathbb{R}^{n\times p}$$

Also for simplicity, let $f(\boldsymbol{\omega}) = \|\mathbf{y}-\mathbf{X}\boldsymbol{\omega}\|_2^2,\ h(\boldsymbol{\omega}) = \lambda \sum_{g\in\mathcal{G}} \|\boldsymbol{\omega_g}\|_2.$

$\mathcal{G}$ is a partition over the whole index set.

e.g., when $p = 10, \mathcal{G}$ can be $\{\{1,2,3\},\{4,5\},\{6,7,8,9,10\}\}$

- For simplicity, we only consider balanced partitions (i.e, the size of each element in *g* is equal).
- Compare over subgradient method (SGD), proximal method (ISTA), and block coordinate descent method (BCD)

Group Lasso – SGD solver

- Calculate the subgradient and iteratively do gradient descent as the Lasso case.

Group Lasso – ISTA solver

We Just need to change Step 2.1, comparing to Lasso.

-Step 2.1 : Solve subproblems for each group g separately as:

For each group $g \in \mathcal{G}$, do

$$\boldsymbol{\omega}^*_{\boldsymbol{g},\boldsymbol{L}} = \mathrm{Prox}_{\frac{\lambda}{L\|\boldsymbol{\omega}\|_2}}(\boldsymbol{\omega}^{\boldsymbol{t}}_{\boldsymbol{g}} - \frac{1}{L}\nabla_{\boldsymbol{\omega}^{\boldsymbol{t}}_{\boldsymbol{g}}} f(\boldsymbol{\omega}^{\boldsymbol{t}}_{\boldsymbol{g}})),$$

Group Lasso – BCD solver

Regular version

For iteration t=1 to $\cdots$

For each group g, iteratively solve the subproblems using SGD

each decent step i:

Norm of this vector

$$\boldsymbol{\omega}_{\boldsymbol{g}}^{(\boldsymbol{t+1})_{\boldsymbol{i}}} = \mathrm{Prox}_{\frac{\lambda\|.\|_2}{L}}\left(\boldsymbol{\omega}_{\boldsymbol{g}}^{(\boldsymbol{t+1})_{\boldsymbol{i-1}}} - \frac{1}{L}\nabla_{\boldsymbol{\omega}_{\boldsymbol{g}}^{(\boldsymbol{t+1})_{\boldsymbol{i-1}}}} f(\boldsymbol{\omega})\right)$$

Stochastic version

For iteration t=1 to $\cdots$

Randomly pick a group g, solve the subproblems using SGD

each decent step i:

$$\boldsymbol{\omega}_{\boldsymbol{g}}^{(\boldsymbol{t+1})_{\boldsymbol{i}}} = \mathrm{Prox}_{\frac{\lambda\|.\|_2}{L}}\left(\boldsymbol{\omega}_{\boldsymbol{g}}^{(\boldsymbol{t+1})_{\boldsymbol{i-1}}} - \frac{1}{L}\nabla_{\boldsymbol{\omega}_{\boldsymbol{g}}^{(\boldsymbol{t+1})_{\boldsymbol{i-1}}}} f(\boldsymbol{\omega})\right)$$

Task 1

Regular task1 : Draw regularization paths.

E.g., let n=50, p=10.

1.Draw $\mathbf{X} \in \mathbb{R}^{n \times p}$, where each element was drawn from $\mathcal{N}(0, \frac{1}{n})$

2.Draw $\boldsymbol{\omega}$ whose sparsity level is $s = 0.5$ where each non-zero element was drawn form $\mathcal{N}(0, 1)$

3.Draw noise vector $\mathbf{m}$ from i.i.d. Gaussian $\mathcal{N}(0, 0.01\|\mathbf{X}\boldsymbol{\omega}\|_2^2/n)$

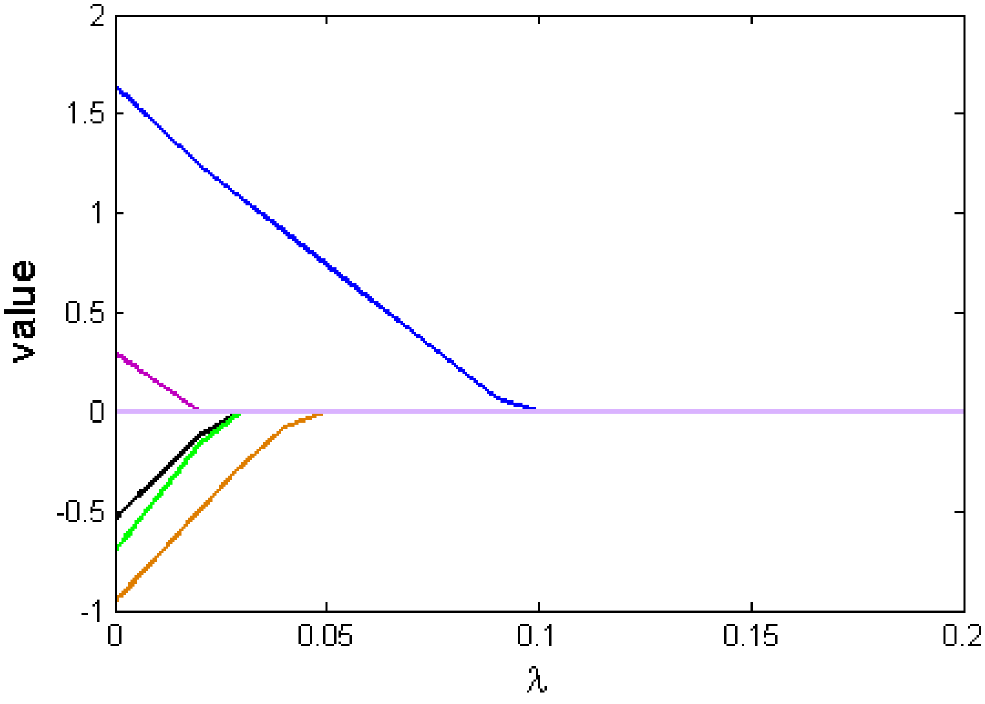
4.Calculate $\mathbf{y} = \mathbf{X}\boldsymbol{\omega} + \mathbf{m}$

5.Solve Lasso on page 3 (any solver is okay).

6.Draw regularization path (variation of each dimension in $\boldsymbol{\omega}$ when tuning λ).

Task1 : Example

The main point here is the piecewise linearity of paths.
Noise is low in this case so you may get perfect result.



Task 2.1

Regular task 2.1 : Solve Lasso with no correlation among features

Let sparsity level $s = \dfrac{\#\text{ of zeros in }\boldsymbol{\omega}}{\text{dimention of }\boldsymbol{\omega}}$

1.Draw $\mathbf{X} \in \mathbb{R}^{n\times p}$, where each element was drawn from $\mathcal{N}(0, \frac{1}{n})$

2.Draw $\boldsymbol{\omega}$ whose sparsity level is s where each non-zero element was drawn form $\mathcal{N}(0,1)$

3.Draw noise vector $\mathbf{m}$ from i.i.d. Gaussian $\mathcal{N}(0, 0.01\|\mathbf{X}\boldsymbol{\omega}\|_2^2/n)$

4.Calculate $\mathbf{y} = \mathbf{X}\boldsymbol{\omega} + \mathbf{m}$

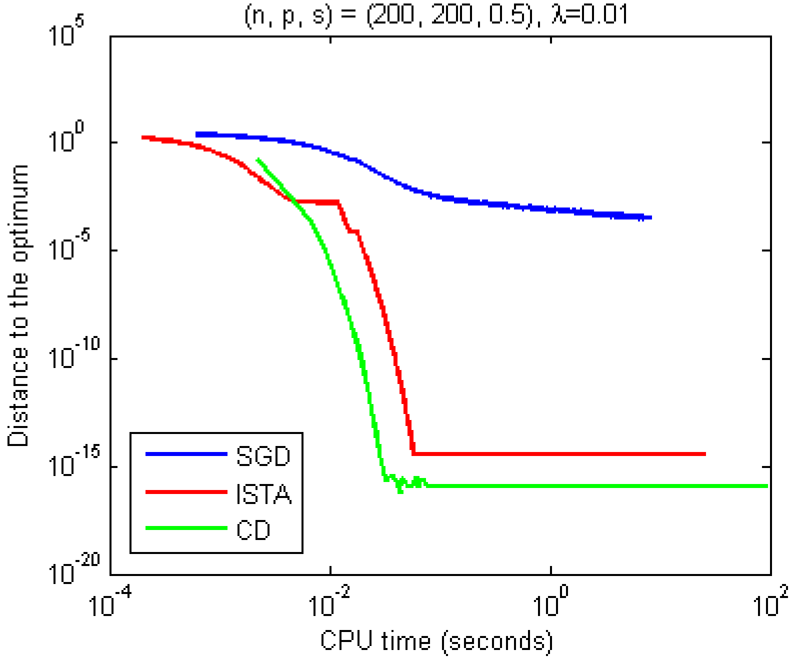
5.Solve Lasso on Page 3, using SGD, ISTA, CD.

6.Plot distance to the optimum objective w.r.t. the CPU time. Explain how you approximate the optimum.

Consider 4 cases : $(n, p, s) = (200, 200, 0, 5), (200, 200, 0.9),$ $(400, 1500, 0.5), (400, 1500, 0.99)$

Task2.1 : Example

- (Example for one case) Your result may not be the same as mine. Do not worry!



Task 2.2

Regular task 2.2 : Solve Lasso **with** correlation among features

I will only show the part different from task 2.1.

1.Draw $\mathbf{X} \in \mathbb{R}^{n \times p}$, where there are correlations among columns.

You may do this via the follwing method.

(1) Draw $\mathbf{X} \in \mathbb{R}^{n \times p}$, where each element was drawn from $\mathcal{N}(0, \frac{1}{n})$

(2) Let $\mathbf{C} = pI + (1-p)E$, where $0 \leq p \leq 1, I$ is an identity matrix

E is a matrix whose elements are all equal to 1.

(3) Do Cholesky decomposition : $\mathbf{C} = \mathbf{D}\mathbf{D}^\top$

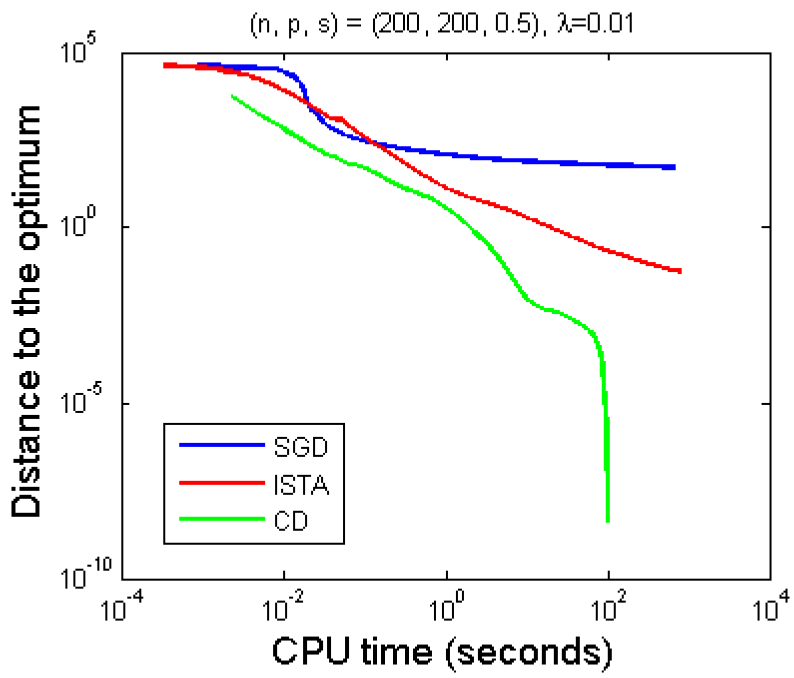
(4) let $\mathbf{X} = \mathbf{D}\mathbf{X}$

Consider 4 cases : $(n, p, s) = (200, 200, 0, 5), (200, 200, 0.9)$,

$(400, 1500, 0.5), (400, 1500, 0.99)$

Task2.2 : Example

- (Example for one case) Your result may not be the same as mine. Do not worry!



Extension Task 3

Extension task 3 : Solve Group Lasso with correlation among features

The correlation matrix is block diagonal (block size = group size).

1.Draw $\mathbf{X} \in \mathbb{R}^{n \times p}$, similar as task 2.2 using block diagonal $\mathbf{C}$.

2.Draw $\boldsymbol{\omega}$ whose sparsity level is s where each non-zero element was drawn form $\mathcal{N}(0, 1)$

3.Draw noise vector $\mathbf{m}$ from i.i.d. Gaussian $\mathcal{N}(0, 0.01\|\mathbf{X}\boldsymbol{\omega}\|_2^2/n)$

4.Calculate $\mathbf{y} = \mathbf{X}\boldsymbol{\omega} + \mathbf{m}$

5.Solve Group Lasso on Page 8.

6.Plot distance to the optimum objective w.r.t. the CPU time.

Let g_0 be the size for each group. Consider 4 cases

$(n, p, s, g_0) = (200, 200, 0,5, 10), (200, 200, 0.5, 50), (200, 200, 0.9, 10),$
$(200, 200, 0.9, 50).$

Tips

- You may consider the following tips when conducting experiments:
  - Tune the hyper-parameter λ to recover the desired result (sparsity pattern in w)
  - Try different strategies in line search (e.g., you may need to tune a and b in the SGD algorithm)
  - Use accurate timers (e.g., 10^{-6} second level accuracy)
  - You may need log-scale to explain the statistics clearly

Bonus

- Here are two problems P1 and P2. Prove that for some T and λ, solving P1 is equivalent to solving P2. You may need to prove the results in two directions.

$$P1: \min_{\boldsymbol{\omega}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\omega}\|_2^2, \text{ s.t. } \|\boldsymbol{\omega}\|_1 \leq T.$$

$$P2: \min_{\boldsymbol{\omega}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\omega}\|_2^2 + \lambda \|\boldsymbol{\omega}\|_1$$

Submission

- Implementation
  - Submit the code implementation before deadline
  - Submit as .zip/.7z/.tar file, including:

    1) Source Code with necessary comments.

    2) Report (.pdf or .doc(x)) containing

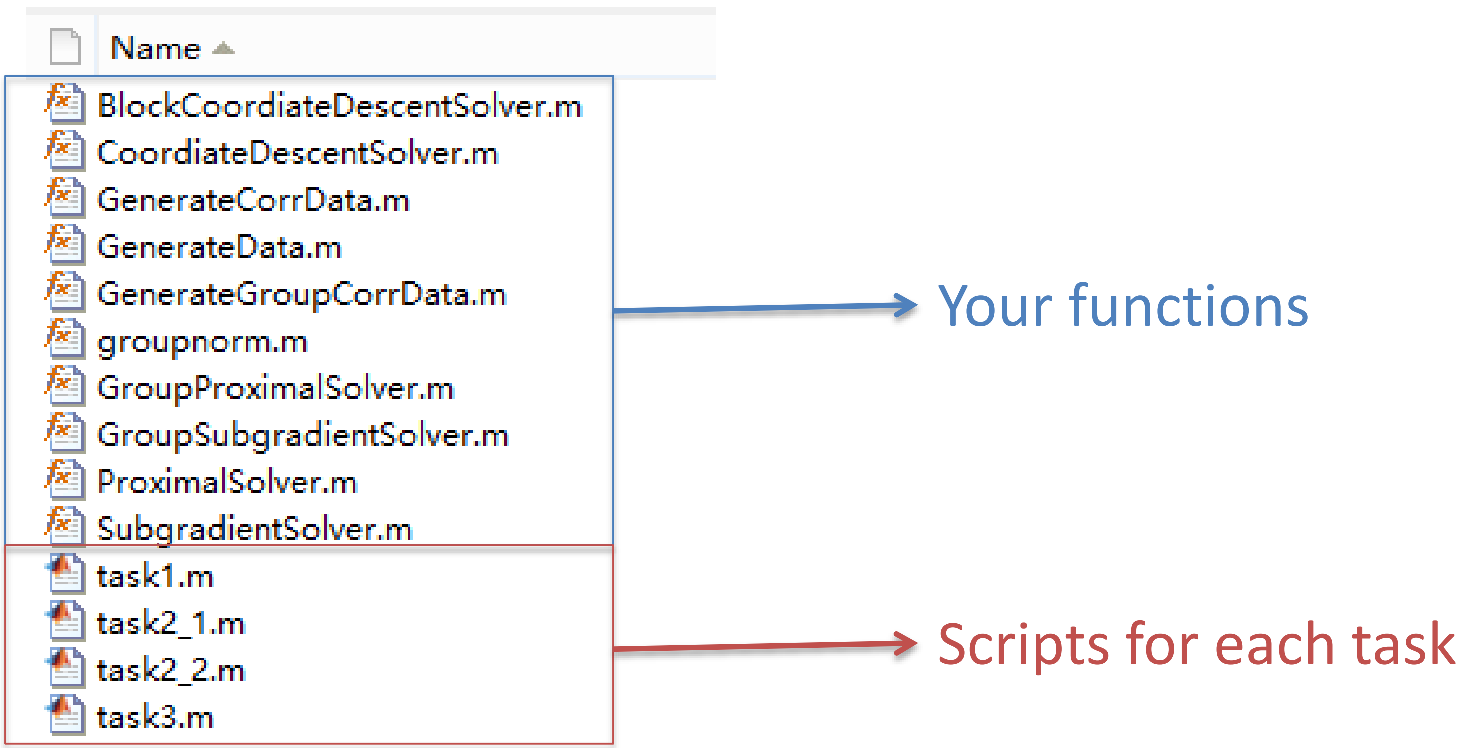
       1. Result for each task.
       2. Explain which algorithm is the fastest in your experiments.
       3. If you do the bonus, contain a readable proof in your report.

    3) ReadMe explaining

       1. How to run your code for each task (you may use a script).
       2. Personal info (name, class, student id, email).

Code Submission Format

- We recommend you pack your code like this, or you may need to explain how to run you code clearly.



Reference

- F. Bach, R. Jenatton, J. Mairal, G. Obozinski. **Optimization with sparsity-inducing penalties**. *Foundations and Trends in Machine Learning,* 4(1):1-106, 2012.

Thank You!