Lasso: Algorithms and Extensions

Yuxin Chen

Princeton University, Spring 2017

Outline

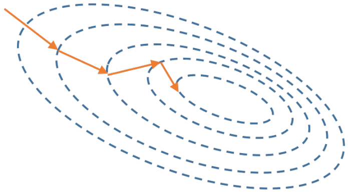
- Proximal operators
- Proximal gradient methods for lasso and its extensions
- Nesterov's accelerated algorithm

Proximal operators

Gradient descent

$$\text{minimize}_{\boldsymbol{\beta} \in \mathbb{R}^p} \quad f(\boldsymbol{\beta})$$

where $f(\boldsymbol{\beta})$ is convex and differentiable



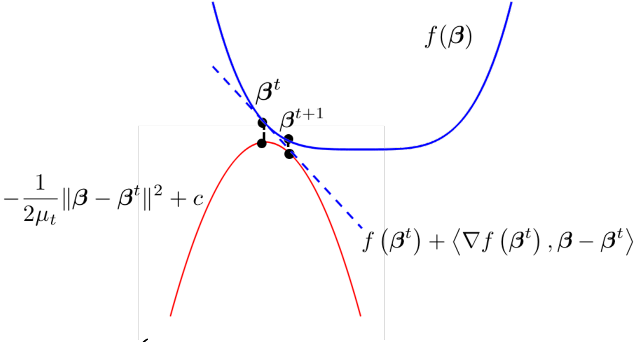
Algorithm 4.1 Gradient descent

for $t = 0, 1, \cdots$**:**

$$\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^t - \mu_t \nabla f(\boldsymbol{\beta}^t)$$

where μ_t: step size / learning rate

A proximal point of view of GD



$$\boldsymbol{\beta}^{t+1}=\arg\min_{\boldsymbol{\beta}}\left\{\underbrace{f(\boldsymbol{\beta}^t)+\langle\nabla f(\boldsymbol{\beta}^t),\boldsymbol{\beta}-\boldsymbol{\beta}^t\rangle}_{\text{linear approximation}}+\underbrace{\frac{1}{2\mu_t}\|\boldsymbol{\beta}-\boldsymbol{\beta}^t\|^2}_{\text{proximal term}}\right\}$$

- When μ_t is small, $\boldsymbol{\beta}^{t+1}$ tends to stay close to $\boldsymbol{\beta}^t$

Proximal operator

If we define the proximal operator

$$\mathsf{prox}_h(\boldsymbol{b}) \;:=\; \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \left\|\boldsymbol{\beta} - \boldsymbol{b}\right\|^2 + h(\boldsymbol{\beta}) \right\}$$

for any convex function h, then one can write

$$\boldsymbol{\beta}^{t+1} = \mathsf{prox}_{\mu_t f_t}\left(\boldsymbol{\beta}^t\right)$$

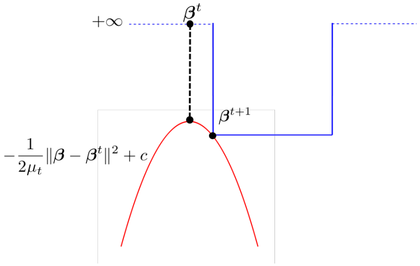
where $f_t(\boldsymbol{\beta}) := f(\boldsymbol{\beta}_t) + \langle \nabla f(\boldsymbol{\beta}_t), \boldsymbol{\beta} - \boldsymbol{\beta}_t \rangle$

Why consider proximal operators?

$$\mathsf{prox}_h(\boldsymbol{b}) \;:=\; \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \left\| \boldsymbol{\beta} - \boldsymbol{b} \right\|^2 + h(\boldsymbol{\beta}) \right\}$$

- It is well-defined under very general conditions (including nonsmooth convex functions)
- The operator can be evaluated efficiently for many widely used functions (in particular, regularizers)
- This abstraction is conceptually and mathematically simple, and covers many well-known optimization algorithms

Example: characteristic functions



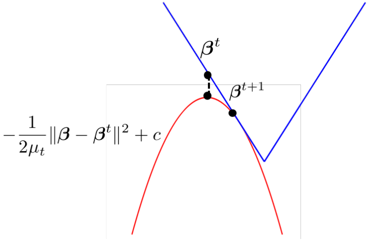
- If h is characteristic function

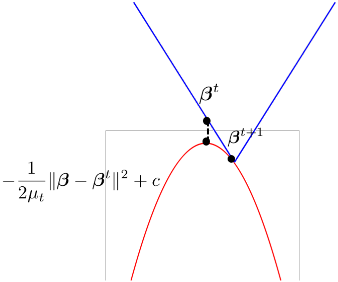
$$h(\boldsymbol{\beta}) = \begin{cases} 0, & \text{if } \boldsymbol{\beta} \in \mathcal{C} \\ \infty, & \text{else} \end{cases}$$

then

$$\mathsf{prox}_h(\boldsymbol{b}) = \arg\min_{\boldsymbol{\beta} \in \mathcal{C}} \|\boldsymbol{\beta} - \boldsymbol{b}\|_2 \quad \text{(Euclidean projection)}$$

Example: ℓ_1 norm





- If $h(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$, then

$$\mathsf{prox}_{\lambda h}(\boldsymbol{b}) = \psi_{\mathrm{st}}(\boldsymbol{b};\lambda)$$

where soft-thresholding $\psi_{\mathrm{st}}(\cdot)$ is applied in an entry-wise manner.

Example: ℓ_2 norm

$$\mathsf{prox}_h(\boldsymbol{b}) \ := \ \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \|\boldsymbol{\beta} - \boldsymbol{b}\|^2 + h(\boldsymbol{\beta}) \right\}$$

- If $h(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|$, then

$$\mathsf{prox}_{\lambda h}(\boldsymbol{b}) = \left(1 - \frac{\lambda}{\|\boldsymbol{b}\|}\right)_+ \boldsymbol{b}$$

where $a_+ := \max\{a, 0\}$. This is called *block soft thresholding*.

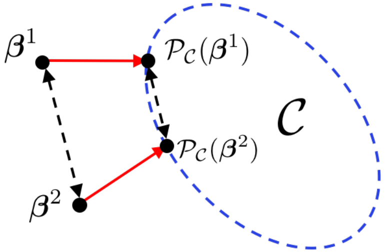
Example: log barrier

$$\mathsf{prox}_h(\boldsymbol{b}) \;:=\; \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \left\| \boldsymbol{\beta} - \boldsymbol{b} \right\|^2 + h(\boldsymbol{\beta}) \right\}$$

- If $h(\boldsymbol{\beta}) = -\sum_{i=1}^{p} \log \beta_i$, then

$$(\mathsf{prox}_{\lambda h}(\boldsymbol{b}))_i = \frac{b_i + \sqrt{b_i^2 + 4\lambda}}{2}$$

Nonexpansiveness of proximal operators

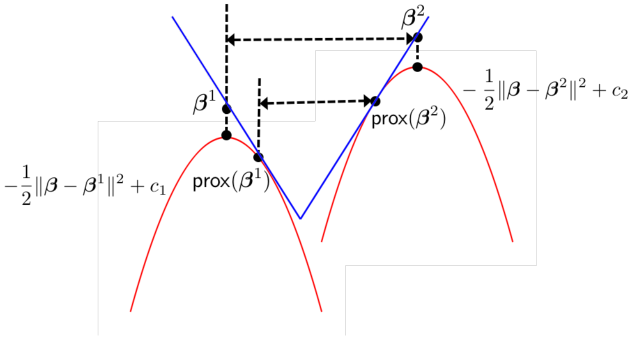


Recall that when $h(\boldsymbol{\beta}) = \begin{cases} 0, & \text{if } \boldsymbol{\beta} \in \mathcal{C} \\ \infty & \text{else} \end{cases}$, $\text{prox}_h(\boldsymbol{\beta})$ is Euclidean projection $\mathcal{P}_{\mathcal{C}}$ onto $\mathcal{C}$, which is nonexpansive:

$$\|\mathcal{P}_{\mathcal{C}}(\boldsymbol{\beta}^1) - \mathcal{P}_{\mathcal{C}}(\boldsymbol{\beta}^2)\| \le \|\boldsymbol{\beta}^1 - \boldsymbol{\beta}^2\|$$

Nonexpansiveness of proximal operators

Nonexpansiveness is a property for general $\text{prox}_h(\cdot)$



Fact 4.1 (Nonexpansiveness)

$$\|\text{prox}_h(\boldsymbol{\beta}^1)-\text{prox}_h(\boldsymbol{\beta}^2)\| \leq \|\boldsymbol{\beta}^1-\boldsymbol{\beta}^2\|$$

- In some sense, proximal operator behaves like projection

Proof of nonexpansiveness

Let $\boldsymbol{z}^1 = \mathsf{prox}_h(\boldsymbol{\beta}^1)$ and $\boldsymbol{z}^2 = \mathsf{prox}_h(\boldsymbol{\beta}^2)$. Subgradient characterizations of $\boldsymbol{z}^1$ and $\boldsymbol{z}^2$ read

$$\boldsymbol{\beta}^1 - \boldsymbol{z}^1 \in \partial h(\boldsymbol{z}^1) \quad \text{and} \quad \boldsymbol{\beta}^2 - \boldsymbol{z}^2 \in \partial h(\boldsymbol{z}^2)$$

The claim would follow if

$$(\boldsymbol{\beta}^1 - \boldsymbol{\beta}^2)^\top (\boldsymbol{z}^1 - \boldsymbol{z}^2) \geq \|\boldsymbol{z}^1 - \boldsymbol{z}^2\|^2 \quad \text{(together with Cauchy-Schwarz)}$$

$$\Longleftarrow \quad (\boldsymbol{\beta}^1 - \boldsymbol{z}^1 - \boldsymbol{\beta}^2 + \boldsymbol{z}^2)^\top (\boldsymbol{z}^1 - \boldsymbol{z}^2) \geq 0$$

$$\Longleftarrow \quad \begin{cases} h(\boldsymbol{z}^2) \geq h(\boldsymbol{z}^1) + \langle \underbrace{\boldsymbol{\beta}^1 - \boldsymbol{z}^1}_{\in \partial h(\boldsymbol{z}^1)},\ \boldsymbol{z}^2 - \boldsymbol{z}^1 \rangle \\ h(\boldsymbol{z}^1) \geq h(\boldsymbol{z}^2) + \langle \underbrace{\boldsymbol{\beta}^2 - \boldsymbol{z}^2}_{\in \partial h(\boldsymbol{z}^2)},\ \boldsymbol{z}^1 - \boldsymbol{z}^2 \rangle \end{cases}$$

Proximal gradient methods

Optimizing composite functions

$$\text{(Lasso)} \quad \text{minimize}_{\boldsymbol{\beta} \in \mathbb{R}^p} \quad \underbrace{\frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|^2}_{:=f(\boldsymbol{\beta})} + \underbrace{\lambda\|\boldsymbol{\beta}\|_1}_{:=g(\boldsymbol{\beta})} = f(\boldsymbol{\beta}) + g(\boldsymbol{\beta})$$

where $f(\boldsymbol{\beta})$ is differentiable, and $g(\boldsymbol{\beta})$ is non-smooth

- Since $g(\boldsymbol{\beta})$ is non-differentiable, we cannot run vanilla gradient descent

Proximal gradient methods

One strategy: replace $f(\boldsymbol{\beta})$ with linear approximation, and compute the proximal solution

$$\boldsymbol{\beta}^{t+1} = \arg\min_{\boldsymbol{\beta}} \left\{ f(\boldsymbol{\beta}^t) + \left\langle \nabla f(\boldsymbol{\beta}^t), \boldsymbol{\beta} - \boldsymbol{\beta}^t \right\rangle + g(\boldsymbol{\beta}) + \frac{1}{2\mu_t} \|\boldsymbol{\beta} - \boldsymbol{\beta}^t\|^2 \right\}$$

The optimality condition reads

$$\mathbf{0} \in \nabla f(\boldsymbol{\beta}^t) + \partial g(\boldsymbol{\beta}^{t+1}) + \frac{1}{\mu_t} \left(\boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^t \right)$$

which is equivalent to optimality condition of

$$\begin{aligned} \boldsymbol{\beta}^{t+1} &= \arg\min_{\boldsymbol{\beta}} \left\{ g(\boldsymbol{\beta}) + \frac{1}{2\mu_t} \left\| \boldsymbol{\beta} - \left(\boldsymbol{\beta}^t - \mu_t \nabla f(\boldsymbol{\beta}^t) \right) \right\|^2 \right\} \\ &= \mathsf{prox}_{\mu_t g} \left(\boldsymbol{\beta}^t - \mu_t \nabla f(\boldsymbol{\beta}^t) \right) \end{aligned}$$

Proximal gradient methods

Alternate between gradient updates on f and proximal minimization on g

Algorithm 4.2 Proximal gradient methods

for $t = 0, 1, \cdots$**:**

$$\boldsymbol{\beta}^{t+1} = \mathsf{prox}_{\mu_t g}\left(\boldsymbol{\beta}^t - \mu_t \nabla f(\boldsymbol{\beta}^t)\right)$$

where μ_t: step size / learning rate

Projected gradient methods

When $g(\boldsymbol{\beta}) = \begin{cases} 0, & \text{if } \boldsymbol{\beta} \in \underbrace{\mathcal{C}}_{\text{convex}} \\ \infty, & \text{else} \end{cases}$ is characteristic function:

$$\begin{aligned} \boldsymbol{\beta}^{t+1} &= \mathcal{P}_{\mathcal{C}}\left(\boldsymbol{\beta}^t - \mu_t \nabla f(\boldsymbol{\beta}^t)\right) \\ &:= \arg\min_{\boldsymbol{\beta} \in \mathcal{C}} \left\| \boldsymbol{\beta} - (\boldsymbol{\beta}^t - \mu_t \nabla f(\boldsymbol{\beta}^t)) \right\| \end{aligned}$$

This is a first-order method to solve the constrained optimization

$$\begin{array}{ll} \text{minimize}_{\boldsymbol{\beta}} & f(\boldsymbol{\beta}) \\ \text{s.t.} & \boldsymbol{\beta} \in \mathcal{C} \end{array}$$

Proximal gradient methods for lasso

For lasso: $f(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|^2$ and $g(\boldsymbol{\beta}) = \lambda\|\boldsymbol{\beta}\|_1$,

$$\begin{aligned} \mathsf{prox}_g(\boldsymbol{\beta}) &= \arg\min_{\boldsymbol{b}} \left\{ \frac{1}{2}\|\boldsymbol{\beta} - \boldsymbol{b}\|^2 + \lambda\|\boldsymbol{b}\|_1 \right\} \\ &= \psi_{\mathrm{st}}\left(\boldsymbol{\beta}; \lambda\right) \end{aligned}$$

$$\Longrightarrow \qquad \boldsymbol{\beta}^{t+1} = \psi_{\mathrm{st}}\left(\boldsymbol{\beta}^t - \mu_t \boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{\beta}^t - \boldsymbol{y});\ \mu_t\lambda\right)$$

(iterative soft thresholding)

Proximal gradient methods for group lasso

Sometimes variables have a natural group structure, and it is desirable to set all variables within a group to be zero (or nonzero) simultaneously

$$\text{(group lasso)} \quad \underbrace{\frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|^2}_{:=f(\boldsymbol{\beta})} + \underbrace{\lambda \sum\nolimits_{j=1}^{k} \|\boldsymbol{\beta}_j\|}_{:=g(\boldsymbol{\beta})}$$

where $\boldsymbol{\beta}_j \in \mathbb{R}^{p/k}$ and $\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_k \end{bmatrix}$.

$$\mathsf{prox}_g(\boldsymbol{\beta}) = \psi_{\mathrm{bst}}\left(\boldsymbol{\beta}; \lambda\right) := \left[\left(1 - \frac{\lambda}{\|\boldsymbol{\beta}_j\|} \right)_+ \boldsymbol{\beta}_j \right]_{1 \le j \le k}$$

$$\Longrightarrow \qquad \boldsymbol{\beta}^{t+1} = \psi_{\mathrm{bst}}\left(\boldsymbol{\beta}^t - \mu_t \boldsymbol{X}^\top (\boldsymbol{X}\boldsymbol{\beta}^t - \boldsymbol{y});\ \mu_t \lambda \right)$$

Proximal gradient methods for elastic net

Lasso does not handle highly correlated variables well: if there is a group of highly correlated variables, lasso often picks one from the group and ignore the rest.

- Sometimes we make a compromise between lasso and ℓ_2 penalties

$$\text{(elastic net)} \quad \underbrace{\frac{1}{2}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|^2}_{:=f(\boldsymbol{\beta})} + \underbrace{\lambda\left\{\|\boldsymbol{\beta}\|_1 + (\gamma/2)\|\boldsymbol{\beta}\|_2^2\right\}}_{:=g(\boldsymbol{\beta})}$$

$$\mathsf{prox}_{\lambda g}(\boldsymbol{\beta}) = \frac{1}{1+\lambda\gamma}\psi_{\mathrm{st}}\left(\boldsymbol{\beta};\lambda\right)$$

$$\Longrightarrow \qquad \boldsymbol{\beta}^{t+1} = \frac{1}{1+\mu_t\lambda\gamma}\psi_{\mathrm{st}}\left(\boldsymbol{\beta}^t - \mu_t\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{\beta}^t - \boldsymbol{y});\ \mu_t\lambda\right)$$

- soft thresholding followed by multiplicative shrinkage

Interpretation: majorization-minimization

$$f_{\mu_t}(\boldsymbol{\beta}, \boldsymbol{\beta}^t) := \underbrace{f(\boldsymbol{\beta}^t) + \left\langle \nabla f(\boldsymbol{\beta}^t), \boldsymbol{\beta} - \boldsymbol{\beta}^t \right\rangle}_{\text{linearization}} + \underbrace{\frac{1}{2\mu_t}\|\boldsymbol{\beta} - \boldsymbol{\beta}^t\|^2}_{\text{trust region penalty}}$$

majorizes $f(\boldsymbol{\beta})$ if $0 < \mu_t < \frac{1}{L}$, where L is Lipschitz constant[1] of $\nabla f(\cdot)$

Proximal gradient descent is a majorization-minimization algorithm

$$\boldsymbol{\beta}^{t+1} = \underbrace{\arg\min_{\boldsymbol{\beta}}}_{\text{minimization}} \Big\{ \underbrace{f_{\mu_t}(\boldsymbol{\beta}, \boldsymbol{\beta}^t) + g(\boldsymbol{\beta})}_{\text{majorization}} \Big\}$$

[1]This means $\|\nabla f(\boldsymbol{\beta}) - \nabla f(\boldsymbol{b})\| \leq L\|\boldsymbol{\beta} - \boldsymbol{b}\|$ for all $\boldsymbol{\beta}$ and $\boldsymbol{b}$

Convergence rate of proximal gradient methods

Theorem 4.2 (fixed step size; Nesterov '07)

Suppose g is convex, and f is differentiable and convex whose gradient has Lipschitz constant L. If $\mu_t \equiv \mu \in (0, 1/L)$, then

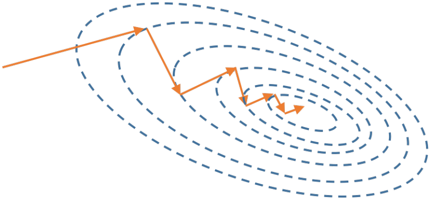
$$f(\boldsymbol{\beta}^t) + g(\boldsymbol{\beta}^t) - \min_{\boldsymbol{\beta}} \{ f(\boldsymbol{\beta}) + g(\boldsymbol{\beta}) \} \;\le\; O\left(\frac{1}{t}\right)$$

- Step size requires an upper bound on L
- May prefer backtracking line search to fixed step size
- **Question:** can we further improve the convergence rate?

Nesterov's accelerated gradient methods

Nesterov's accelerated method

Problem of gradient descent: zigzagging



Nesterov's idea: include a momentum term to avoid overshooting

Nesterov's accelerated method

Nesterov's idea: include a momentum term to avoid overshooting

$$
\begin{aligned}
\boldsymbol{\beta}^t &= \mathsf{prox}_{\mu_t g}\left(\boldsymbol{b}^{t-1} - \mu_t \nabla f\left(\boldsymbol{b}^{t-1}\right)\right) \\
\boldsymbol{b}^t &= \boldsymbol{\beta}^t + \underbrace{\alpha_t\left(\boldsymbol{\beta}^t - \boldsymbol{\beta}^{t-1}\right)}_{\text{momentum term}} \qquad \text{(extrapolation)}
\end{aligned}
$$

- A simple (but mysterious) choice of extrapolation parameter

$$\alpha_t = \frac{t-1}{t+2}$$

- Fixed size $\mu_t \equiv \mu \in (0, 1/L)$ or backtracking line search
- Same computational cost per iteration as proximal gradient

Convergence rate of Nesterov's accelerated method

Theorem 4.3 (Nesterov '83, Nesterov '07)

Suppose f is differentiable and convex and g is convex. If one takes $\alpha_t = \frac{t-1}{t+2}$ and a fixed step size $\mu_t \equiv \mu \in (0, 1/L)$, then

$$f(\boldsymbol{\beta}^t) + g(\boldsymbol{\beta}^t) - \min_{\boldsymbol{\beta}} \left\{ f(\boldsymbol{\beta}) + g(\boldsymbol{\beta}) \right\} \;\leq\; O\left(\frac{1}{t^2}\right)$$

In general, this rate cannot be improved if one only uses gradient information!

Numerical experiments (for lasso)

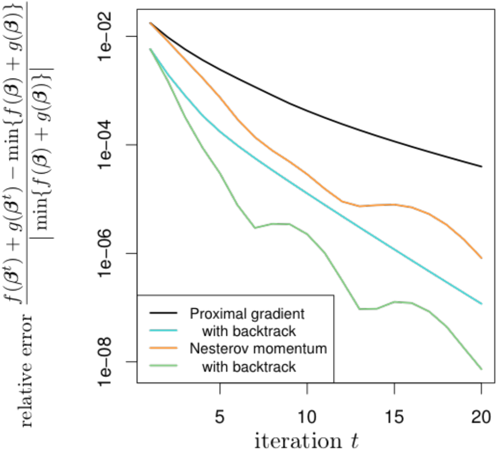


Figure credit: Hastie, Tibshirani, & Wainwright '15

Reference

[1] "*Proximal algorithms*," Neal Parikh and S. Boyd, *Foundations and Trends in Optimization*, 2013.

[2] "*Convex optimization algorithms*," D. Bertsekas, *Athena Scientific*, 2015.

[3] "*Convex optimization: algorithms and complexity*," S. Bubeck, *Foundations and Trends in Machine Learning*, 2015.

[4] "*Statistical learning with sparsity: the Lasso and generalizations*," T. Hastie, R. Tibshirani, and M. Wainwright, 2015.

[5] "*Model selection and estimation in regression with grouped variables*," M. Yuan and Y. Lin, *Journal of the royal statistical society*, 2006.

[6] "*A method of solving a convex programming problem with convergence rate* $O(1/k^2)$," Y. Nestrove, *Soviet Mathematics Doklady*, 1983.

Reference

[7] "*Gradient methods for minimizing composite functions*,", Y. Nesterov, *Technical Report*, 2007.

[8] "*A fast iterative shrinkage-thresholding algorithm for linear inverse problems*," A. Beck and M. Teboulle, *SIAM journal on imaging sciences*, 2009.