ECE 18-898G: Special Topics in Signal Processing: Sparsity, Structure, and Inference

Sparse Recovery using L1 minimization - algorithms

Yuejie Chi

Department of Electrical and Computer Engineering

Carnegie Mellon University

Spring 2018

Outline

- Lasso with orthogonal design
- Proximal operators
- Proximal gradient methods for lasso and its extensions
- Nesterov's accelerated algorithm (FISTA)

These are useful in general for compound optimization problems.

Sparse recovery by ℓ_0 regularization

As a warm up, consider a sparsifying basis $\boldsymbol{X} \in \mathbb{R}^{n \times n}$ that is orthonormal, we wish to solve the following sparsity-promoting problem regularized by ℓ_0 norm:

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \quad \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \underbrace{\lambda\|\boldsymbol{\beta}\|_0}_{\text{penalized by sparsity level}} \tag{4.1}$$

- The first term is the approximation error: $\|\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\|^2$.
- The second term is the model complexity: $\|\hat{\boldsymbol{\beta}}\|_0$.

We will discuss "regularized" algorithms throughout this lecture.

Orthogonal design

Since $\boldsymbol{X}$ is orthonormal,

$$\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 = \|\boldsymbol{X}^\top \boldsymbol{y} - \boldsymbol{\beta}\|^2.$$

Without loss of generality, suppose $\boldsymbol{X} = \boldsymbol{I}$, then (4.1) reduces to

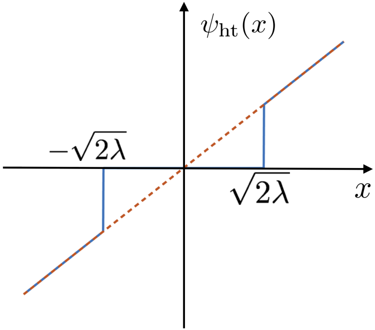
$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \quad \sum_{i=1}^{n} \frac{1}{2} \left[(y_i - \beta_i)^2 + \lambda \cdot \mathbf{1}\{\beta_i \neq 0\} \right]$$

Solving this problem gives

$$\hat{\beta}_i = \begin{cases} 0, & |y_i| \leq \sqrt{2\lambda} \\ y_i, & |y_i| > \sqrt{2\lambda} \end{cases} \qquad \text{hard thresholding}$$

- Keep large coefficients; discard small coefficients

The case $\boldsymbol{X} = \boldsymbol{I}$



$$\hat{\beta}_i = \psi_{\mathsf{ht}}(y_i; \sqrt{2\lambda}) := \begin{cases} 0, & |y_i| \le \sqrt{2\lambda} \\ y_i, & |y_i| > \sqrt{2\lambda} \end{cases}$$

hard thresholding

Hard thresholding preserves data outside threshold zone

Convex relaxation: Lasso (Tibshirani ’96)

Lasso (Least absolute shrinkage and selection operator)

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \quad \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1 \tag{4.2}$$

for some regularization parameter $\lambda > 0$.

- It is equivalent to

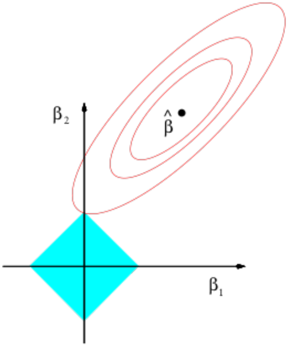
  $$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \quad \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 \quad \text{s.t.} \qquad \|\boldsymbol{\beta}\|_1 \le t$$

  for some t that depends on λ (no explicit formula)
    - a quadratic program (QP) with convex constraints
- λ controls model complexity: larger λ restricts the parameters more; smaller λ frees up more parameters
- Also related to Basis Pursuit:

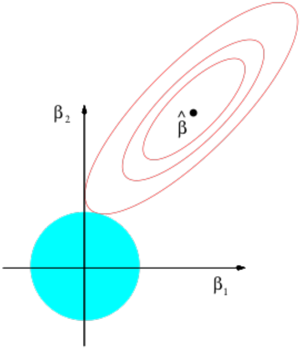
  $$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \quad \|\boldsymbol{\beta}\|_1 \quad \text{s.t.} \qquad \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\| \le \epsilon$$

Lasso vs. ridge regression

$\hat{\boldsymbol{\beta}}$: least squares solution



$$\begin{aligned} \text{minimize}_{\boldsymbol{\beta}} \quad & \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\| \\ \text{s.t.} \quad & \|\boldsymbol{\beta}\|_1 \le t \end{aligned}$$

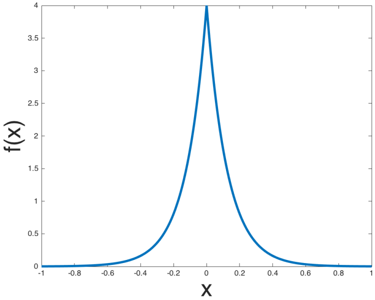


$$\begin{aligned} \text{minimize}_{\boldsymbol{\beta}} \quad & \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\| \\ \text{s.t.} \quad & \|\boldsymbol{\beta}\|_2 \le t \end{aligned}$$

Fig. credit: Hastie, Tibshirani, & Wainwright

A Bayesian interpretation

Orthogonal design: $\boldsymbol{y} = \boldsymbol{\beta} + \boldsymbol{\eta}$ with $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$.



Impose an i.i.d. prior on β_i to encourage sparsity *(Gaussian is not a good choice)*:

$$\text{(Laplacian prior)} \qquad \mathbb{P}(\beta_i = z) = \frac{\lambda}{2} e^{-\lambda |z|}$$

The tail decays exponentially.

A Bayesian interpretation of Lasso

Posterior of $\boldsymbol{\beta}$:

$$\mathbb{P}(\boldsymbol{\beta} \mid \boldsymbol{y}) \propto \mathbb{P}(\boldsymbol{y}|\boldsymbol{\beta})\mathbb{P}(\boldsymbol{\beta}) \propto \prod_{i=1}^{n} e^{-\frac{(y_i-\beta_i)^2}{2\sigma^2}} \frac{\lambda}{2} e^{-\lambda|\beta_i|}$$

$$\propto \prod_{i=1}^{n} \exp\left\{ -\frac{(y_i-\beta_i)^2}{2\sigma^2} - \lambda|\beta_i| \right\}$$

$\Longrightarrow$ maximum *a posteriori* (MAP) estimator:

$$\arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left\{ \frac{(y_i-\beta_i)^2}{2\sigma^2} + \lambda|\beta_i| \right\} \quad \text{(Lasso)}$$

Implication: Lasso is MAP estimator under Laplacian prior

Example: orthogonal design

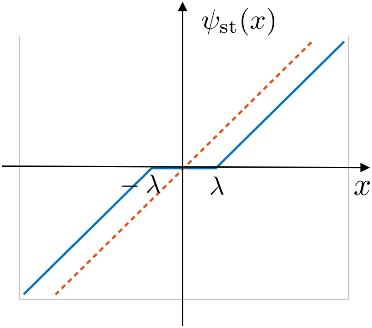
Suppose $\boldsymbol{X} = \boldsymbol{I}$, then Lasso reduces to

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \quad \sum_{i=1}^{n} \left[\frac{1}{2}(y_i - \beta_i)^2 + \lambda |\beta_i| \right]$$

The Lasso estimate $\hat{\boldsymbol{\beta}}$ is then given by

$$\hat{\beta}_i = \begin{cases} y_i - \lambda, & y_i \geq \lambda \\ y_i + \lambda, & y_i \leq -\lambda \\ 0, & \text{else} \end{cases} \qquad \text{soft thresholding}$$

Example: orthogonal design



$$\hat{\beta}_i = \psi_{\mathsf{st}}(y_i; \lambda) = \begin{cases} y_i - \lambda, & y_i \geq \lambda \\ y_i + \lambda, & y_i \leq -\lambda \\ 0, & \text{else} \end{cases}$$ soft thresholding

Soft thresholding shrinks data towards 0 outside threshold zone

Optimality condition for convex functions

For any convex function $f(\boldsymbol{\beta})$, $\boldsymbol{\beta}^*$ is an optimal solution iff $\mathbf{0} \in \partial f(\boldsymbol{\beta}^*)$, where $\partial f(\boldsymbol{\beta})$ is the set of all subgradients at $\boldsymbol{\beta}$

- The subgradient of $f(\beta) = \frac{1}{2}(y - \beta)^2 + \lambda|\beta|$ can be written as

$$g = \beta - y + \lambda s$$

with s is a subgradient of $f(\beta) = |\beta|$ if

$$\begin{cases} s = \operatorname{sign}(\beta), & \text{if } \beta \neq 0 \\ s \in [-1, 1], & \text{if } \beta = 0 \end{cases} \tag{4.3}$$

- We see that $\hat{\beta} = \psi_{\mathrm{st}}(y; \lambda)$ by checking optimality conditions for two cases:
  - If $|y| \leq \lambda$, taking $\beta = 0$ and $s = y/\lambda$ gives $g = 0$
  - If $|y| > \lambda$, taking $\beta = y - \operatorname{sign}(y)\lambda$ gives $g = 0$

Solving LASSO in general cases

Composite optimization problems

General composite optimization problem:

$$\hat{\boldsymbol{\beta}} = \mathsf{argmin}_{\boldsymbol{\beta}} \left\{ F(\boldsymbol{\beta}) = f(\boldsymbol{\beta}) + g(\boldsymbol{\beta}) \right\}$$

- $f(\boldsymbol{\beta})$ is convex and differentiable, e.g. approximation error
- $g(\boldsymbol{\beta})$ is convex, possibly non-differentiable, e.g. regularizers

Examples:

- LASSO: $f(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2$, and $g(\boldsymbol{\beta}) = \lambda\|\boldsymbol{\beta}\|_1$.
- Matrix completion:

$$f(\boldsymbol{X}) = \|\mathcal{P}_\Omega(\boldsymbol{Y} - \boldsymbol{X})\|_{\mathsf{F}}^2, \qquad g(\boldsymbol{X}) = \lambda\|\boldsymbol{X}\|_*$$

where $\|\boldsymbol{X}\|_*$ is the nuclear norm.

Motivation

Standard methods (e.g. subgradient methods) for solving composite optimization has very slow convergence rate.

We would discuss accelerated proximal gradient methods that

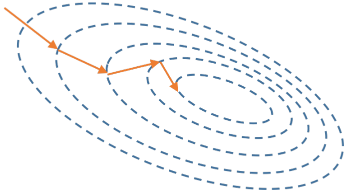
- is iterative, and has low computational cost (first-order algorithm, which requires computation of a single gradient per iteration);
- has quadratic convergence rate;
- performs well in practice and works for a large class of problems.

Proximal gradient methods

Gradient descent

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

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



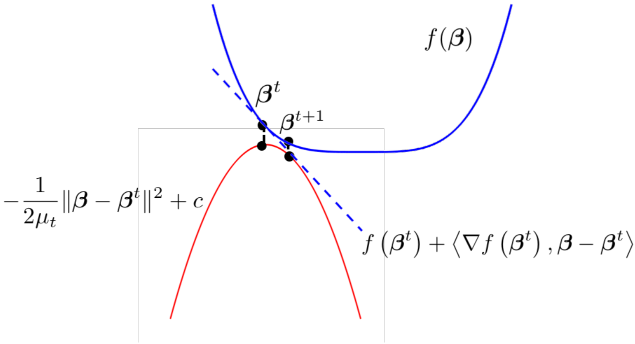
Algorithm 4.1 Gradient descent

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



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

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

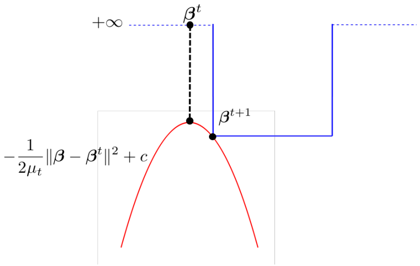
Gradient descent is performing proximal mapping at every iteration.

Why consider proximal operators?

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

- 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



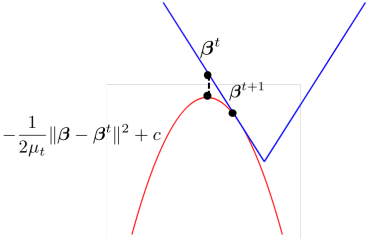
- If h is characteristic function

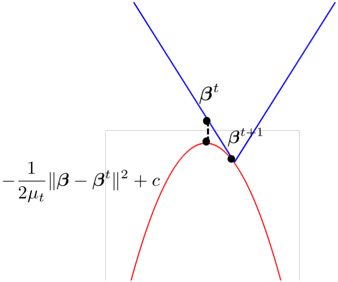
$$h(\boldsymbol{\beta}) = \begin{cases} 0, & \text{if } \boldsymbol{\beta} \in \mathcal{C} \\ \infty, & \text{else} \end{cases}$$

then

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

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

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

- 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*.

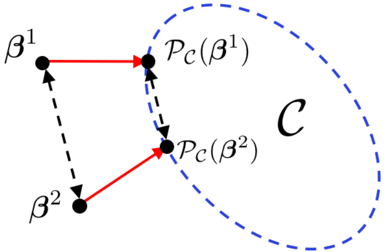
Example: log barrier

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

- 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

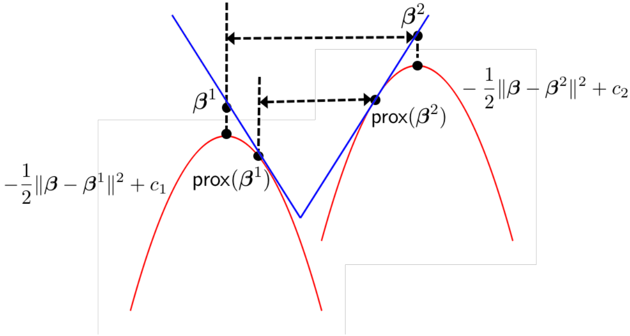


Recall that when $h(\boldsymbol{\beta}) = \begin{cases} 0, & \text{if } \boldsymbol{\beta} \in \mathcal{C} \\ \infty & \text{else} \end{cases}$, $\text{prox}_h(\boldsymbol{\beta})$ is Euclidean projection $\mathcal{P}_\mathcal{C}$ onto $\mathcal{C}$, which is nonexpansive:

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

Nonexpansiveness of proximal operators

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



Fact 4.1 (Nonexpansiveness)

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

- 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 mapping 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:

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

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

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

Proximal gradient methods for lasso

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

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

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

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 $\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_k \end{bmatrix}$.

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

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

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

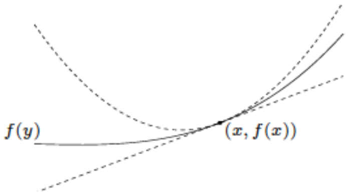
$$\boldsymbol{\beta}^{t+1} = \underbrace{\arg\min_{\boldsymbol{\beta}}}_{\text{minimization}} \Big\{ \underbrace{f_{\mu_t}(\boldsymbol{\beta}, \boldsymbol{\beta}^t) + g(\boldsymbol{\beta})}_{\text{majorization}} \Big\}$$

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

Assumptions for convergence

- $g : \mathbb{R}^n \mapsto \mathbb{R}$ is a continuous convex function, possibly nonsmooth;
- $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a smooth convex function that is continuously differentiable with *Lipschitz constant*:

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$



$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{L_f}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2, \quad \forall \boldsymbol{x}, \boldsymbol{y}$$

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

$$F(\boldsymbol{\beta}^t) - F(\hat{\boldsymbol{\beta}}) \;\leq\; O\left(\frac{\|\boldsymbol{\beta}^0 - \hat{\boldsymbol{\beta}}\|^2}{t\mu}\right)$$

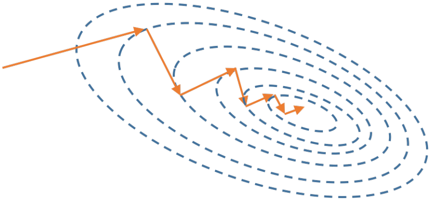
- Step size requires an upper bound on L
  - For LASSO problems, we have $L = \sigma_{\max}(\boldsymbol{A}^\top \boldsymbol{A})$.
- May prefer backtracking line search to fixed step size
- **Question:** can we further improve the convergence rate?

Nesterov's accelerated gradient methods

- We will first examine Nesterov's acceleration method (1983) for smooth convex functions;
- We then extend it to optimizing composite functions, using FISTA (Beck and Teboulle, 2009), which extends Nesterov's method to proximal gradient methods.

Nesterov's accelerated method

Problem of gradient descent: zigzagging



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

Accelerated Gradient descent

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

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

Algorithm 4.3 Accelerated Gradient descent

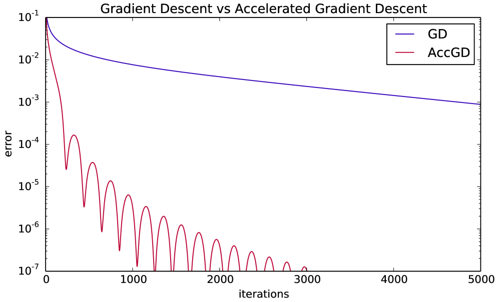
for $t = 0, 1, \cdots$**:**

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

$$\boldsymbol{b}^t = \boldsymbol{\beta}^t + \underbrace{\alpha_t \left(\boldsymbol{\beta}^t - \boldsymbol{\beta}^{t-1}\right)}_{\text{momentum term}}$$

where μ_t: step size / learning rate, α_t the the extrapolation parametre

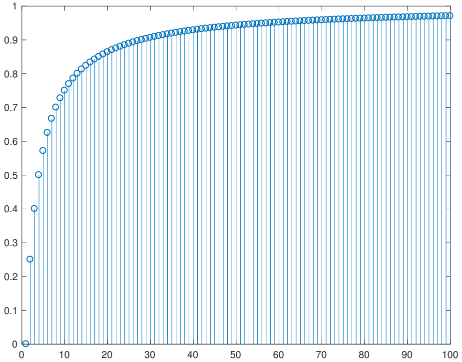
With/Without Acceleration



Momentum parameter

A simple (but mysterious) choice of extrapolation parameter

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



Another choice: let $s_1 = 1$, $s_{t+1} = \frac{1+\sqrt{1+4s_t^2}}{2}$, and $\alpha_t = \frac{s_t - 1}{s_{t+1}}$.

Accelerated proximal method (FISTA)

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

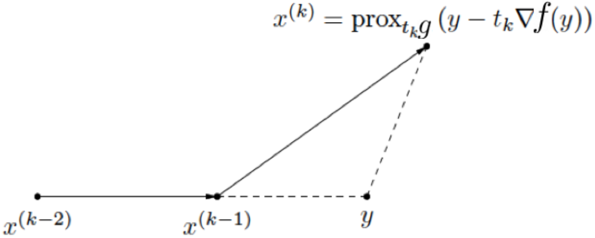
$$\begin{aligned}\boldsymbol{\beta}^t &= \mathsf{prox}_{\mu_t g}\left(\boldsymbol{b}^{t-1} - \mu_t \nabla f\left(\boldsymbol{b}^{t-1}\right)\right) \\ \boldsymbol{b}^t &= \boldsymbol{\beta}^t + \underbrace{\alpha_t\left(\boldsymbol{\beta}^t - \boldsymbol{\beta}^{t-1}\right)}_{\text{momentum term}} \qquad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

- 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

Interpretation



Convergence of FISTA

Theorem 4.3 (Nesterov '83, Nesterov '07, Beck & M. Teboulle '09)

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

$$F(\boldsymbol{\beta}^t) - F(\hat{\boldsymbol{\beta}}) \;\le\; O\left(\frac{1}{t^2}\right)$$

- Improves upon $O(\frac{1}{t})$ convergence than proximal gradient method.
- in general un-improvable

Numerical experiments (for lasso)

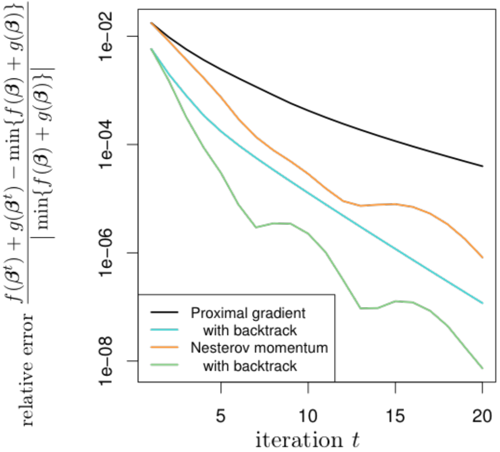


Figure credit: Hastie, Tibshirani, & Wainwright '15

Computational-Statistical Trade-off

If there is indeed a ground truth $\boldsymbol{\beta}^\star$ and we wish $\hat{\boldsymbol{\beta}}$ is close to $\boldsymbol{\beta}^\star$; we have a sequence of $\{\boldsymbol{\beta}_t\}$ and hope $\boldsymbol{\beta}_t$ converges to $\hat{\boldsymbol{\beta}}$. At a fixed t, we may bound

$$\|\boldsymbol{\beta}_t - \boldsymbol{\beta}^\star\|_2 \leq \underbrace{\|\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}\|_2}_{\text{computational error}} + \underbrace{\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\|_2}_{\text{statistical error}}$$

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. Nesterov, *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.