Iterative Optimization Algorithms

The Big Picture

- Most of the optimization problems in this course don't have closed form solutions. Ridge regression is an exception.
- In these cases we resort to iterative optimization algorithms. Even when a closed form solution is available, an iterative solver can be more computationally efficient.
- Today we'll overview several iterative solvers

Outline

- Gradient descent
- Stochastic gradient descent
- The subgradient method
- Coordinate descent

ERM

- *(Regularized) empirical risk minimization* learns f by solving

$$\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(\boldsymbol{x}_i)) + \lambda\Omega(f),$$

- Different choices of $L, \mathcal{F}, \Omega$ give rise to different methods.
- Example: ridge regression without offset

$$\min_{\boldsymbol{w}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^T\boldsymbol{x}_i)^2 + \lambda\|\boldsymbol{w}\|^2,$$

Computational Complexity

- What is the computational complexity of computing

$$\widehat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda n \boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

where

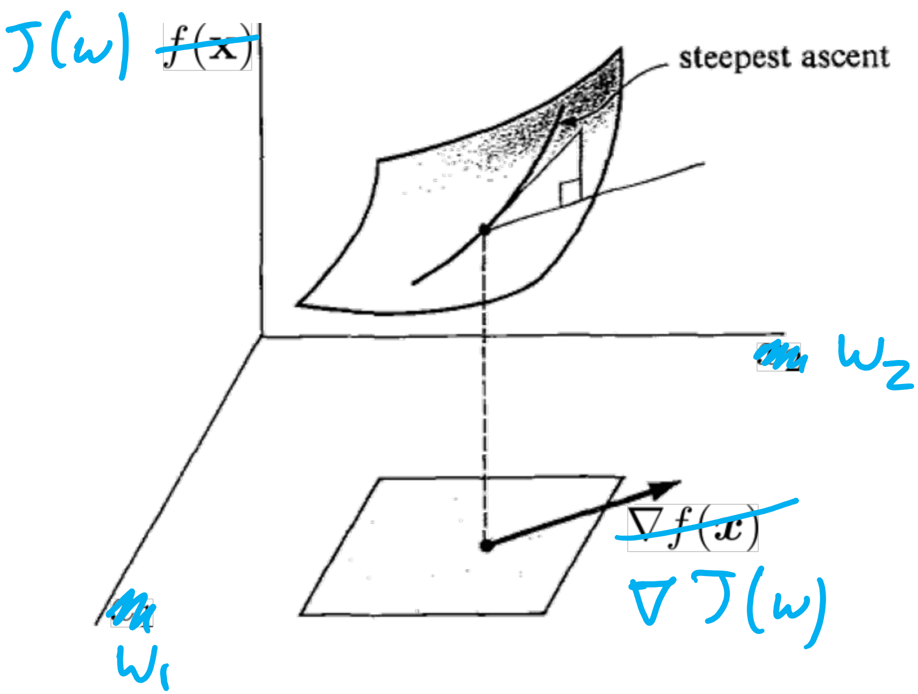
$$X = \underbrace{\begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}}_{(n \times d)}, \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$(d \times d)$ $X^T X$: $O(nd^2)$ Overall: $O(nd^2 + d^3)$

$X^T X$: $O(d^3)$

Gradient

The gradient of a function is a vector that points in the direction of steepest ascent



$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

Gradient Descent

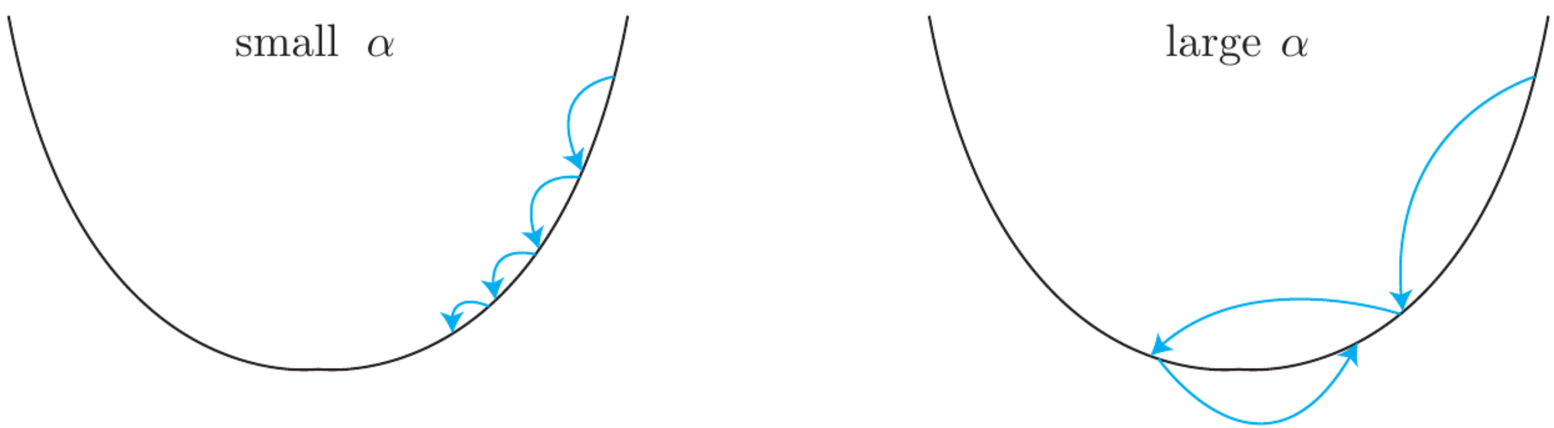
- Consider minimizing the generic objective function $J(\boldsymbol{\theta})$
- Initial guess $\boldsymbol{\theta}_0$
- For $t = 1, \ldots,$ `max_iter`

$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \alpha_t \nabla J(\boldsymbol{\theta}_{t-1})$$

If convergence condition satisfied, exit

End

step size,
learning rate



Gradient Descent for Linear Regr.

The regularized least-squares (i.e., ridge regression) objective function can be written ~~(after eliminating b)~~

$$J(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{A}\boldsymbol{w} + \boldsymbol{r}^T\boldsymbol{w} + c,$$

where $\boldsymbol{A} = 2(\boldsymbol{X}^T\boldsymbol{X} + n\lambda\boldsymbol{I})$, $\boldsymbol{r} = -2\boldsymbol{X}^T\boldsymbol{y}$, and $c = \boldsymbol{y}^T\boldsymbol{y}$

1. $\nabla J(\boldsymbol{w}) =$ $Aw + r = 2(X^TX + n\lambda I)w - 2X^Ty$
2. What is the computational complexity of gradient descent in terms of d and n?

$$X^TXw = \underbrace{X^T\underbrace{(Xw)}_{O(nd)}}_{O(nd)}$$

Conclusion:

$O(nd)$ operations per iteration of GD.

conjugate gradient descent

Stochastic Gradient Descent

- Suppose it is possible to write

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} J_i(\theta)$$

where $J_i(\boldsymbol{\theta})$ depends on the training data only through $(\boldsymbol{x}_i, y_i)$ ~~or, in the case of ridge regression, $(\tilde{\boldsymbol{x}}_i, \tilde{y}_i)$~~.

- *Stochastic gradient descent* is the following variation on gradient descent:

- Initialize $\boldsymbol{\theta}_0$, set $t = 0$

- For $j = 1, \ldots,$ `max_iter`

  For $i = 1, \ldots, n$ *in random order*

  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \alpha_j \nabla J_i(\boldsymbol{\theta}_{t-1})$

  $t \leftarrow t + 1$

  End

  If convergence condition satisfied, exit

  End

$$\nabla J(\theta) = \sum \nabla J_i(\theta)$$

$$\underbrace{\begin{array}{l} g = 0 \\ \text{For } i = 1 \text{ to } n \\ \quad g = g + \nabla J_i(\theta_{t-1}) \\ \text{End} \\ \theta_t = \theta_{t-1} - \alpha g \end{array}}_{\text{GD}}$$

SGD for Ridge Regression

$$J(w) = \sum_{i=1}^{n} (y_i - w^T x_i)^2 + n\lambda \|w\|^2$$

$$= \sum_{i=1}^{n} J_i(w)$$

where $J_i(w) = (y_i - w^T x_i)^2 + \lambda \|w\|^2$

$$\nabla J_i(w) = -2(y_i - w^T x_i) x_i + 2\lambda w \in \mathbb{R}^d$$

$O(d)$ operations per update

Poll

True or False: If the step-size α_j is carefully chosen, then the ridge regression objective function decreases at every iteration j of gradient descent (unless you're already at a local min)

(A) True ✓

(B) False

line search
backtracking

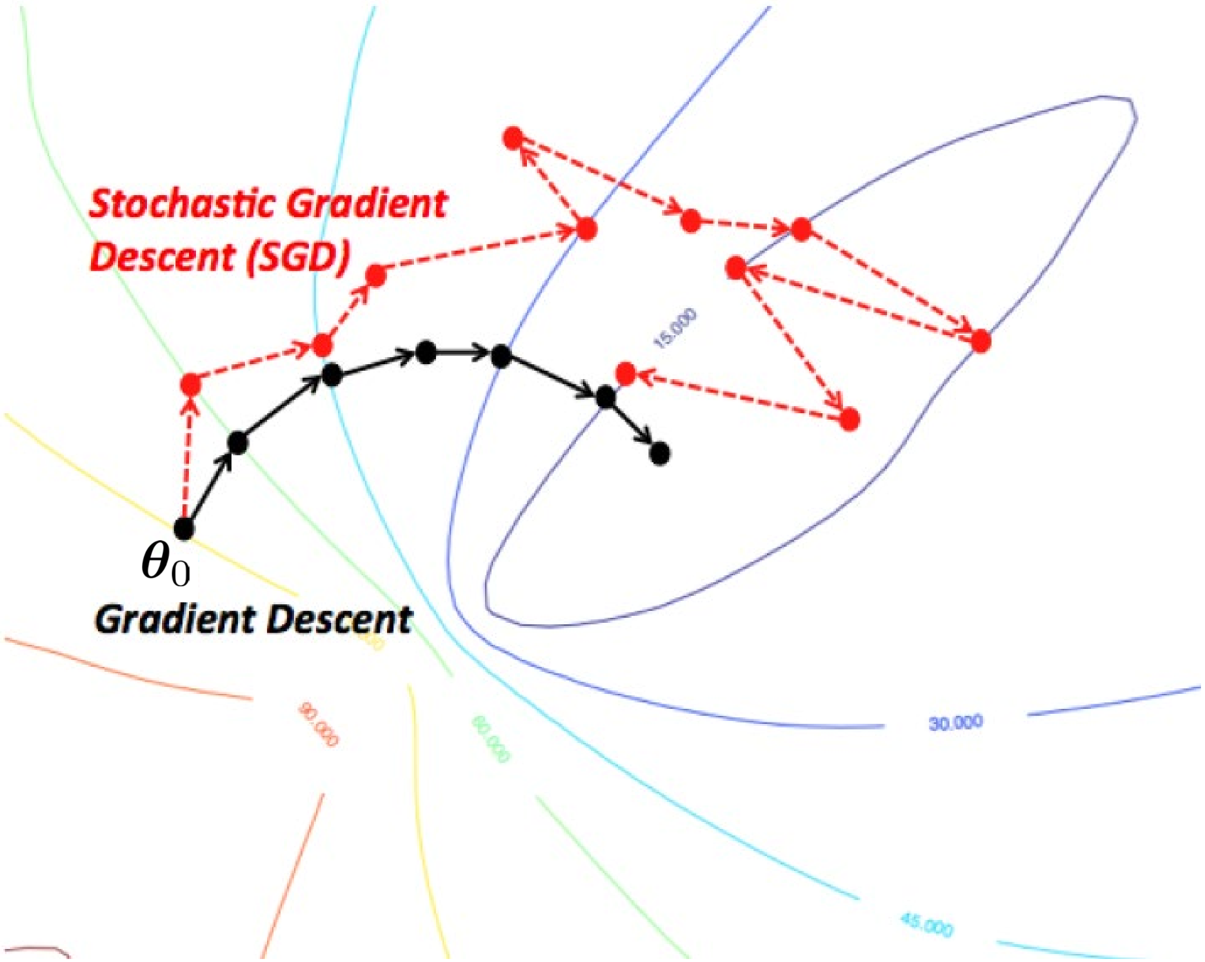
Poll

True or False: If the step-size α_j is carefully chosen, then the ridge regression objective function decreases at every iteration of stochastic gradient descent

(A) True

(B) False

GD vs SGD



Stochastic Gradient Descent (SGD)
θ_0
Gradient Descent
15,000
30,000
45,000
60,000
90,000

The Lasso

- How can we solve

$$\min_{\boldsymbol{w},b} \frac{1}{n} \sum_{i=1}^{n} (y_i - \boldsymbol{w}^T \boldsymbol{x}_i - b)^2 + \lambda \|\boldsymbol{w}\|_1?$$

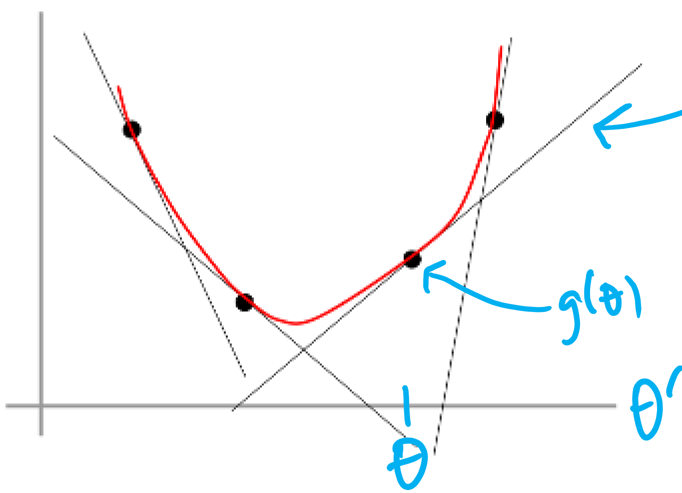
Subgradient Method~~s~~

- The *subgradient method* is a generalization of gradient descent that applies to *nondifferentiable, convex* objective functions, like the lasso or ERM with hinge loss

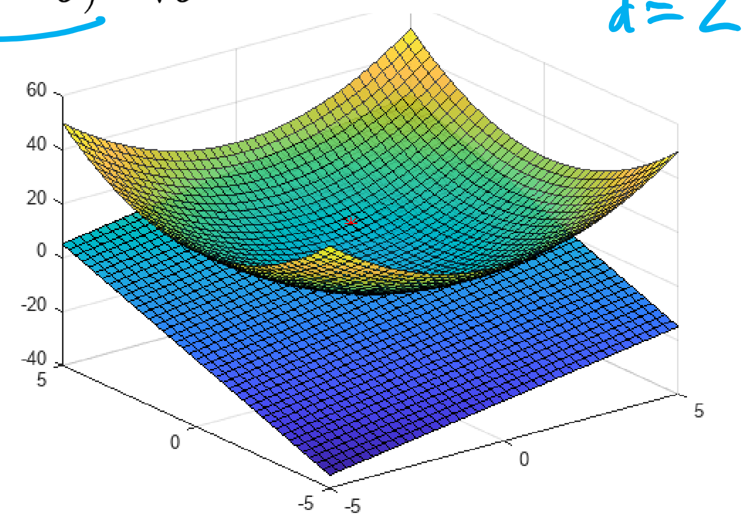
- Let $g : \mathbb{R}^d \to \mathbb{R}$ be convex, and let $\boldsymbol{\theta} \in \mathbb{R}^d$. If g is differentiable, then $\boldsymbol{u} = \nabla g(\boldsymbol{\theta})$ is the only vector such that

$$g(\boldsymbol{\theta}') \geq \underbrace{g(\boldsymbol{\theta}) + \boldsymbol{u}^T(\boldsymbol{\theta}' - \boldsymbol{\theta})} \quad \forall \boldsymbol{\theta}'$$

d=1



d=2

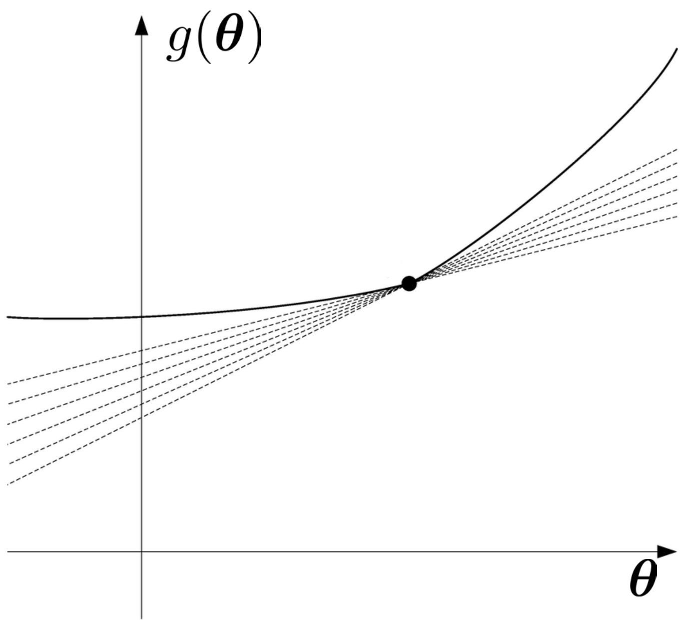


https://mathoverflow.net/questions/327940/existence-of-a-strictly-convex-function-interpolating-given-gradients-and-values

https://www.mathworks.com/help/matlab/math/calculate-tangent-plane-to-surface.html

Subgradients

- If g is convex but *not* differentiable, then for some $\boldsymbol{\theta}$, there may be many $\boldsymbol{u}$ satisfying the previous inequality.
- We define the *subdifferential* of g at $\boldsymbol{\theta}$, denoted $\boxed{\partial g(\boldsymbol{\theta}),}$ to be the set of all $\boldsymbol{u}$ satisfying the inequality.
- A *subgradient* is any element of the subdifferential.
- In the figure, the subdifferential is the interval $\left[g'_{-}(\theta), g'_{+}(\theta)\right]$ where $g'_{-}(\theta), g'_{+}(\theta)$ denote the left and right derivatives.



Subgradient Method

- In the *subgradient method*, we update the parameter just as in gradient descent, but where the gradient is replaced by *any* subgradient.

- Pseudo-code for minimizing $g(\boldsymbol{\theta})$:

  - initialize $\boldsymbol{\theta}_0$
  - $t \leftarrow 0$
  - Repeat
    - select $\boldsymbol{u}_t \in \partial g(\boldsymbol{\theta}_t)$
    - $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \alpha_t \boldsymbol{u}_t$
    - $t \leftarrow t + 1$
  - Until stopping criterion satisfied

- If it is possible to write

$$g(\boldsymbol{\theta}) = \sum_{i=1}^{n} g_i(\theta)$$

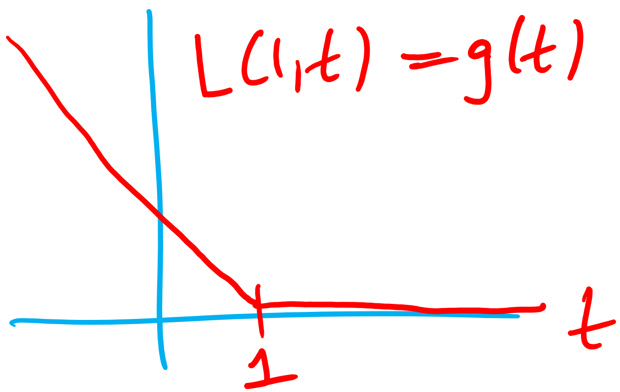
  then we can also have a *stochastic subgradient method*, analogous to SGD.

Subgradients

- What is the subdifferential of the hinge loss?

$L(y,t) = \max(0, 1-yt)$

Take $y=1$ for concreteness



$L(1,t) = g(t)$

$$\partial g(t) = \begin{cases} -1 & t < 1 \\ [-1, 0] & t = 1 \\ 0 & t > 1 \end{cases}$$

Poll

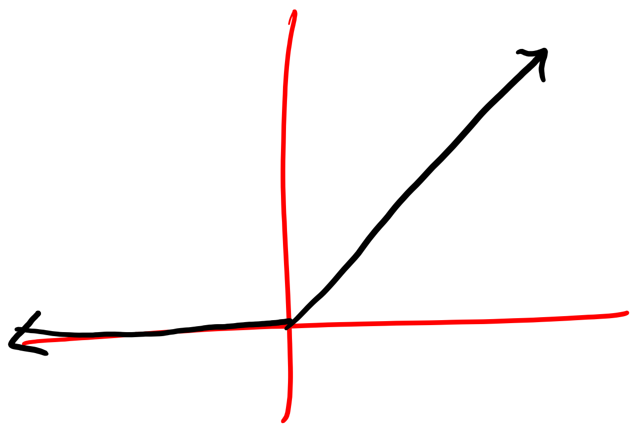
- The ReLU activation function is defined by $\sigma(t) = \max(0, t)$. The subdifferential of the ReLU function at $t = 0$ is

  (A) $[-1, 0]$

  (B) $[0, 1]$ ✓

  (C) $[-1, 1]$

  (D) None of the above



Subgradients

$$w = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$$

- What is a the subdifferential of $\|\boldsymbol{w}\|_1$?

$$\|w\|_1 = \sum_{j=1}^{d} |w_j|$$

$$\partial_{w_j} \|w\|_1 = \partial |w_j| = \begin{cases} -1 & w_j < 0 \\ [-1, 1] & w_j = 0 \\ 1 & w_j > 0 \end{cases}$$

$$\partial_w \|w\|_1 = \begin{bmatrix} \mathrm{sign}(w_1) \\ \vdots \\ \mathrm{sign}(w_d) \end{bmatrix} \quad \text{where } \mathrm{sign}(0) = [-1, 1]$$

https://see.stanford.edu/materials/lsocoee364b/01-subgradients_notes.pdf
https://www.seas.ucla.edu/~vandenbe/236C/lectures/subgradients.pdf

Coordinate Descent

$$\theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_p \end{bmatrix}$$

- The subgradient method can converge slowly for the lasso.
- Coordinate descent cycles through the coordinates of $\boldsymbol{\theta}$, updating one coordinate while leaving the others fixed.
- If J is the objective function, and $\boldsymbol{\theta}^{(0)}$ is the initial iterate, then $\boldsymbol{\theta}^{(1)}$ is obtained by

$$\theta_1^{(1)} = \arg\min_{\phi \in \mathbb{R}} J(\phi, \theta_2^{(0)}, \theta_3^{(0)}, \ldots, \theta_p^{(0)})$$

$$\theta_2^{(1)} = \arg\min_{\phi \in \mathbb{R}} J(\theta_1^{(1)}, \phi, \theta_3^{(0)}, \ldots, \theta_p^{(0)})$$

$$\vdots$$

Coordinate Descent

- In general, if $\boldsymbol{\theta}^{(t)}$ is the t^{th} iterate, then

$$\theta_j^{(t)} = \arg\min_{\cancel{\theta}\ \phi} J\left(\theta_1^{(t)}, \ldots, \theta_{j-1}^{(t)}, \overset{\phi}{\cancel{\theta}}, \theta_{j+1}^{(t-1)}, \ldots, \theta_p^{(t-1)}\right)$$

Apply to lasso w/ $\theta = \begin{bmatrix} b \\ w \end{bmatrix} \in \mathbb{R}^{d+1}$

Coordinate Descent for the Lasso

- Let's apply CD to minimize

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^T\boldsymbol{x}_i - b)^2 + \lambda\|\boldsymbol{w}\|_1$$

- b update: $b^{(t)} = \bar{y} - (w^{(t-1)})^T \bar{x}$

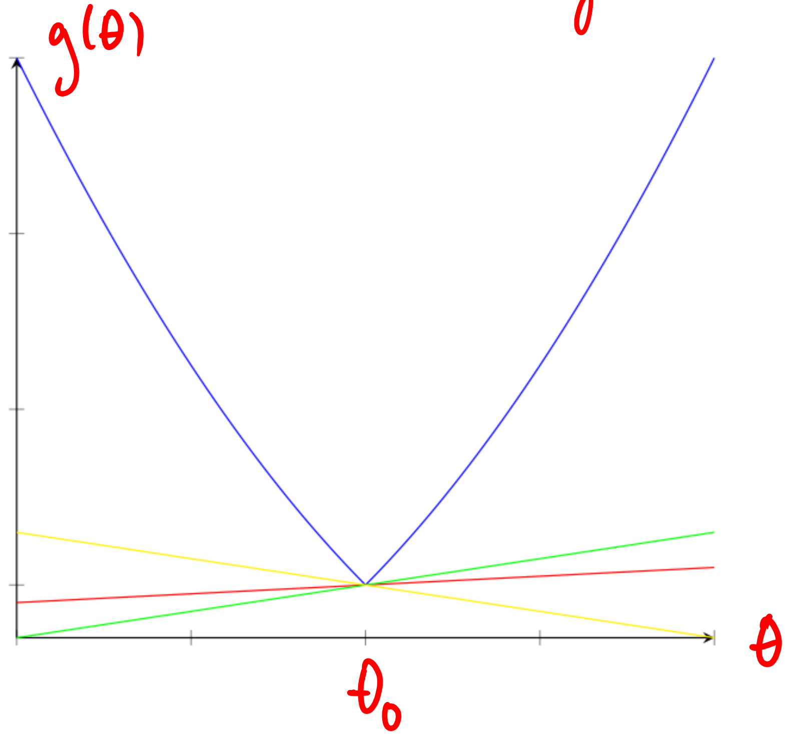
- To update $w_j^{(t)}$ we need to solve

$$\min_{w_j} \; g(w_j) := \frac{1}{n}\sum_{i=1}^{n}\left(y_i - [w_1^{(t)} \cdots w_{j-1}^{(t)} \; w_j \; w_{j+1}^{(t-1)} \cdots w_d^{(j-1)}] \begin{bmatrix} x_{i1} \\ \vdots \\ x_{id} \end{bmatrix} - b^{(t)}\right)^2 + \lambda|w_j|$$

- How to solve?

More Sugradients

- Let $g(\boldsymbol{\theta})$ be a convex function. Then a point $\boldsymbol{\theta}_0$ is a global minimizer of g if and only if $0 \in \partial g(\theta)$. This follows directly from the definitions of global min and subdifferential.



Lasso Subproblem Solution

- Introduce the following notation:

$$\boldsymbol{w}_{-j}^{(t)} = \begin{bmatrix} w_1^{(t)} \\ \vdots \\ w_{j-1}^{(t)} \\ w_{j+1}^{(t-1)} \\ \vdots \\ w_d^{(t-1)} \end{bmatrix} \qquad \boldsymbol{x}_{i,-j} = \begin{bmatrix} x_{i,1} \\ \vdots \\ x_{i,j-1} \\ x_{i,j+1} \\ \vdots \\ x_{i,d} \end{bmatrix}.$$

- You will show on HW4 that the subdifferential of $g(w_j)$ is

$$\partial g(w_j) = \begin{cases} a_j^{(t)} w_j - c_j^{(t)} - \lambda, & w_j < 0 \\ [a_j^{(t)} w_j - c_j^{(t)} - \lambda, a_j^{(t)} w_j - c_j^{(t)} + \lambda], & w_j = 0 \\ a_j^{(t)} w_j - c_j^{(t)} + \lambda, & w_j > 0 \end{cases}.$$

where

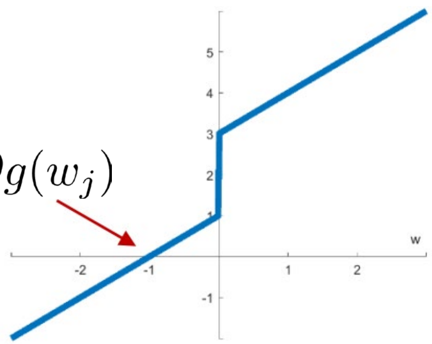
$$a_j^{(t)} = \frac{2}{n} \sum_i x_{ij}^2, \qquad c_j^{(t)} = \frac{2}{n} \sum_i x_{ij} (y_i - (\boldsymbol{w}_{-j}^{(t)})^T \boldsymbol{x}_{i,-j} - b^{(t)}).$$

Lasso Subproblem Solution

- On HW4, you will show that there is a unique value of w_j such that $0 \in \partial g(w_j)$.
- There are three cases to consider, shown below.

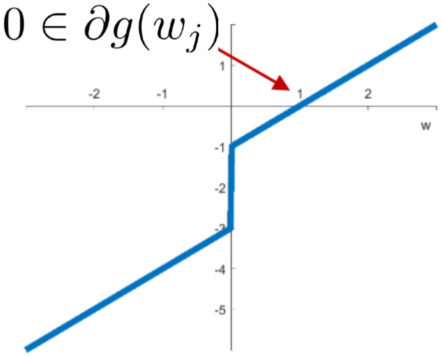
$c_j^{(t)} < -\lambda$

$0 \in \partial g(w_j)$



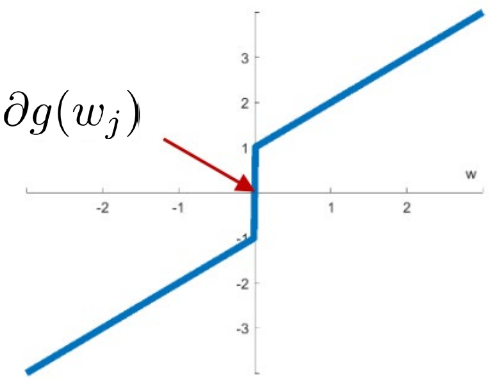
$c_j^{(t)} > \lambda$

$0 \in \partial g(w_j)$



$c_j^{(t)} \in [-\lambda, \lambda]$

$0 \in \partial g(w_j)$



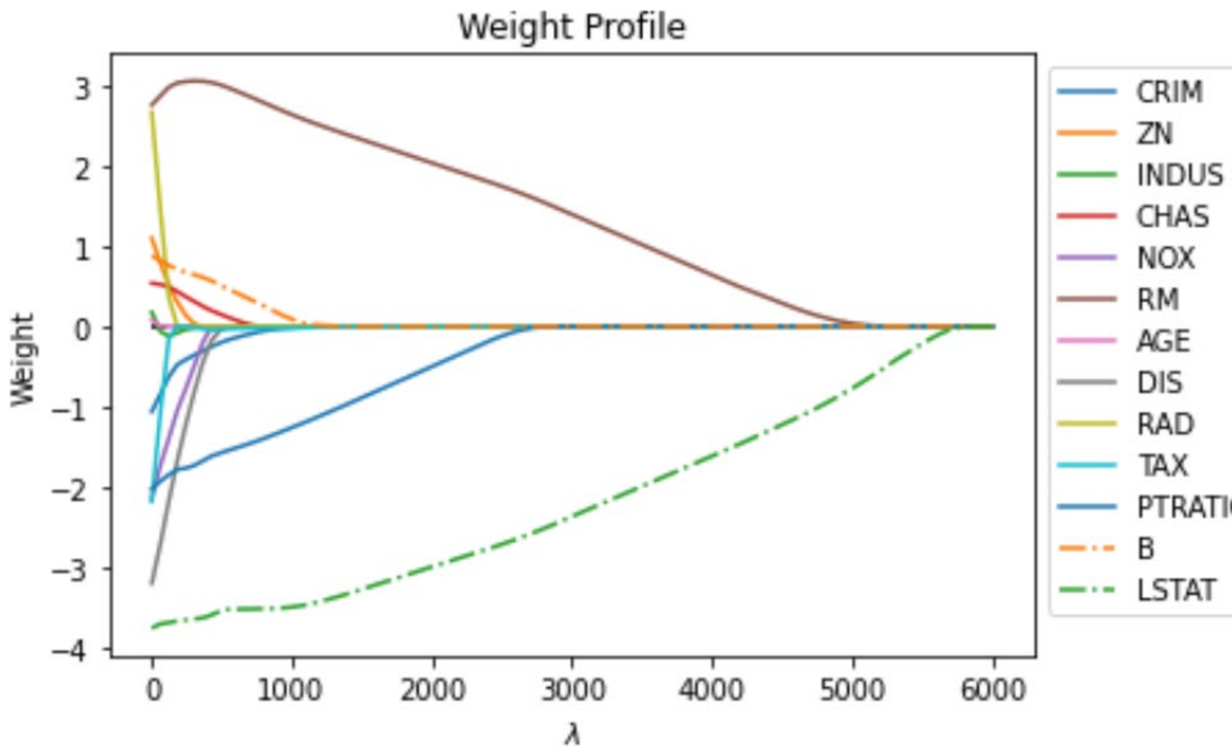
CD for Lasso: Final Algorithm

- Initialize $\boldsymbol{w}^{(0)}, b^{(0)}$
- For $t = 1, 2, \ldots$
  - $b^{(t)} = \bar{y} - (\boldsymbol{w}^{(t-1)})^T \bar{\boldsymbol{x}}$
  - For $j = 1, \ldots, d$
    - Compute $c_j^{(t)}, a_j^{(t)}$
    - $w_j^{(t)} = \text{soft}(c_j^{(t)}/a_j^{(t)}, \lambda/a_j^{(t)})$
  - If stopping criterion met, break

Soft thresholding:

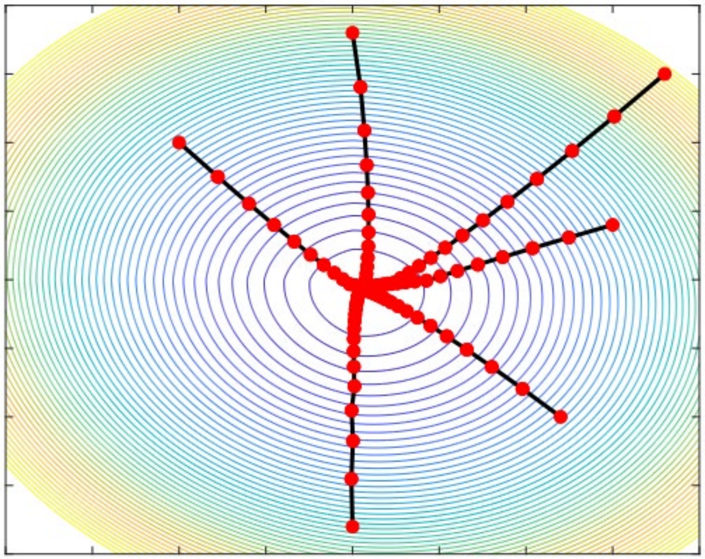
$$\text{soft}(\alpha, \beta) := \begin{cases} \alpha - \beta & \text{if } \alpha > \beta \\ 0 & \text{if } \alpha \in [-\beta, \beta] \\ \alpha + \beta & \text{if } \alpha < -\beta \end{cases}$$

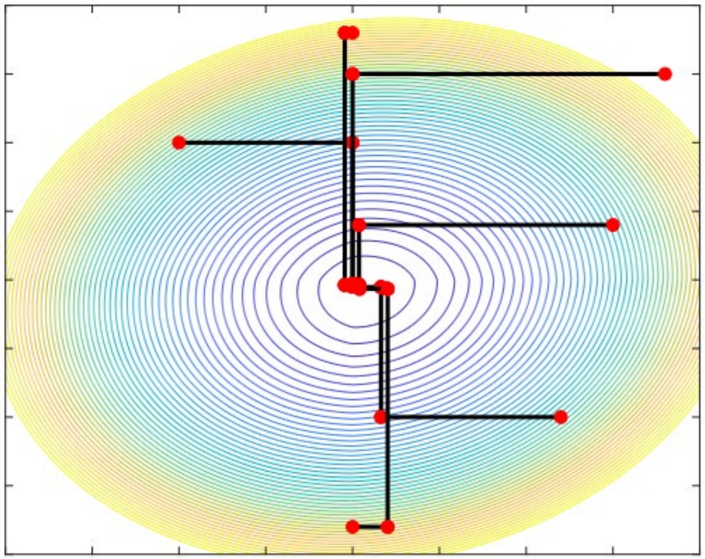
Boston Housing Data: Solution Path



Subgradient Method vs CD

- $y = 1 \cdot x - 1 + z$
- x, z are independent $\mathcal{N}(0, 1)$
- $\lambda = 100$
- $\eta = 0.1$





Final Thoughts on Lasso

- CD is typically much faster to converge that the subgradient method
- CD also has no tuning parameters
- Least angle regression (LARS): Solve Lasso for all λ in one algorithm
- Selected features are not stable: slight perturbation of training data can lead to significant changes to sparsity pattern

List of Methods

- Gradient descent / SGD
- Subgradient method / SSM
- Coordinate descent
- Newton’s method
- Quasi-Newton methods
- Majorize-minimize
- ADMM
- . . .

- Convergence guarantees exist for all of these methods, but are beyond our time constraints