NYU Center for Data Science

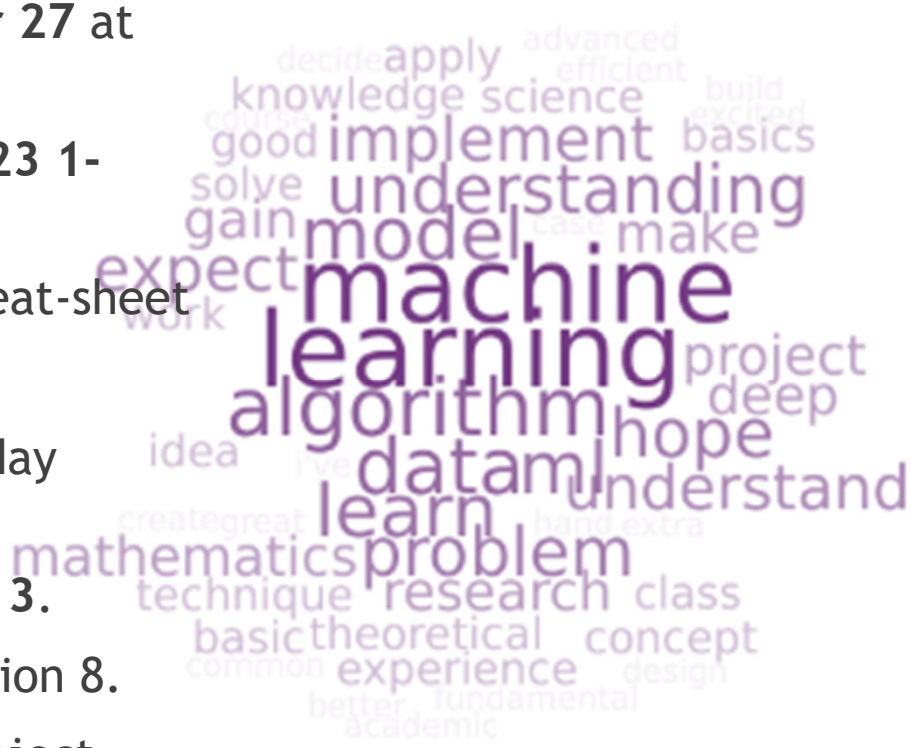
DS-GA 3001.007
Introduction to Machine Learning

Lecture 6

Regularization – Algorithms for Ridge and Lasso

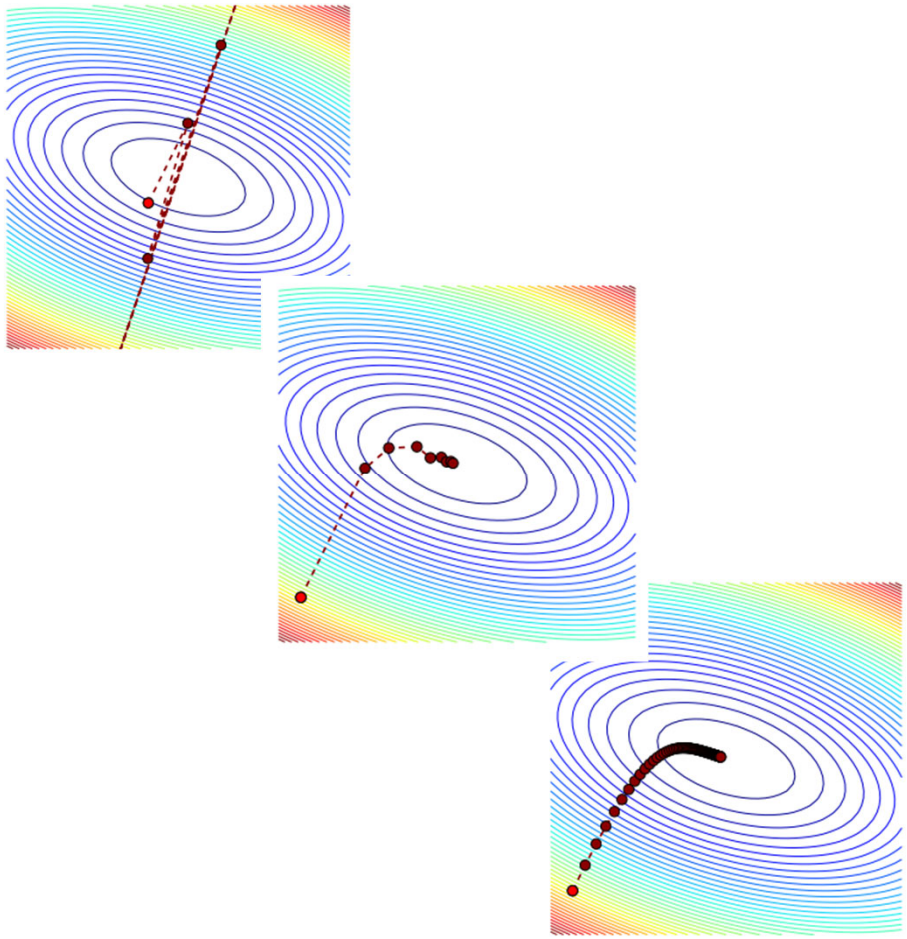
Announcements

- Homework 3 extended to **Sunday October 27** at 11:59pm
- Midterm in class on **Wednesday October 23 1-2:40pm**.
  - Exam will be pencil and paper with cheat-sheet
  - Section 7 will review relevant material
  - Practice exam will be posted on Thursday October 17.
- Project 1 Proposal due **Sunday November 3**.
  - We will discuss sample projects in Section 8.
  - Please post to the Forum about the project.



Review

- Fixed Step Size
  - Learning rate 1/(max of derivative)
  - Number Iterations O(1/error)
- Varying Step Size
  - Back-tracking Line Search
    - Number Iterations O(1/error) with better constant
  - Strongly Convex Functions
    - Learning rate 1/(iteration)
    - Number Iterations O(log(1/error))



Bound on Derivative

Definition (Lipschitz continuity). *A function* $f : \mathbb{R}^n \to \mathbb{R}^m$ *is Lipschitz continuous with Lipschitz constant* L *if for any* $\vec{x}, \vec{y} \in \mathbb{R}^n$

$$||f(\vec{y}) - f(\vec{x})||_2 \leq L ||\vec{y} - \vec{x}||_2 .$$

Bound on Derivative

Definition (Lipschitz continuity). *A function* $f : \mathbb{R}^n \to \mathbb{R}^m$ *is Lipschitz continuous with Lipschitz constant* L *if for any* $\vec{x}, \vec{y} \in \mathbb{R}^n$

$$||f(\vec{y}) - f(\vec{x})||_2 \leq L ||\vec{y} - \vec{x}||_2 .$$

small input small output stable

Theorem *If the gradient of a function* $f : \mathbb{R}^n \to \mathbb{R}$ *is Lipschitz continuous with Lipschitz constant* L,

$$||\nabla f(\vec{y}) - \nabla f(\vec{x})||_2 \leq L ||\vec{y} - \vec{x}||_2$$

then for any $\vec{x}, \vec{y} \in \mathbb{R}^n$

$$f(\vec{y}) \leq f(\vec{x}) + \nabla f(\vec{x})^T (\vec{y} - \vec{x}) + \frac{L}{2} ||\vec{y} - \vec{x}||_2^2 .$$

Gradient Descent Decreases Function

Corollary . *Let $\vec{x}^{(i)}$ be the ith iteration of gradient descent and $\alpha_i \geq 0$ the ith step size, if ∇f is L-Lipschitz continuous,*

$$f\left(\vec{x}^{(k+1)}\right) \leq f\left(\vec{x}^{(k)}\right) - \alpha_k \left(1 - \frac{\alpha_k L}{2}\right) \left|\left|\nabla f\left(\vec{x}^{(k)}\right)\right|\right|_2^2.$$

Gradient Descent Decreases Function

Corollary *Let $\vec{x}^{(i)}$ be the ith iteration of gradient descent and $\alpha_i \geq 0$ the ith step size, if ∇f is L-Lipschitz continuous,*

$$f\left(\vec{x}^{(k+1)}\right) \leq f\left(\vec{x}^{(k)}\right) - \alpha_k \left(1 - \frac{\alpha_k L}{2}\right) \left|\left|\nabla f\left(\vec{x}^{(k)}\right)\right|\right|_2^2.$$

Proof. Applying the quadratic upper bound we obtain

$$f\left(\vec{x}^{(k+1)}\right) \leq f\left(\vec{x}^{(k)}\right) + \nabla f\left(\vec{x}^{(k)}\right)^T \left(\vec{x}^{(k+1)} - \vec{x}^{(k)}\right) + \frac{L}{2} \left|\left|\vec{x}^{(k+1)} - \vec{x}^{(k)}\right|\right|_2^2.$$

L is from second dreiv

Gradient Descent Decreases Function

Corollary *Let $\vec{x}^{(i)}$ be the ith iteration of gradient descent and $\alpha_i \geq 0$ the ith step size, if ∇f is L-Lipschitz continuous,*

as much as possible

$$f\left(\vec{x}^{(k+1)}\right) \leq f\left(\vec{x}^{(k)}\right) - \alpha_k \left(1 - \frac{\alpha_k L}{2}\right) \left|\left|\nabla f\left(\vec{x}^{(k)}\right)\right|\right|_2^2.$$

d/d lapha k () = 1-alpha k L = 0
alpha k = 1/L

Proof. Applying the quadratic upper bound we obtain

$$f\left(\vec{x}^{(k+1)}\right) \leq f\left(\vec{x}^{(k)}\right) + \nabla f\left(\vec{x}^{(k)}\right)^T \left(\vec{x}^{(k+1)} - \vec{x}^{(k)}\right) + \frac{L}{2} \left|\left|\vec{x}^{(k+1)} - \vec{x}^{(k)}\right|\right|_2^2.$$

The result follows because $\vec{x}^{(k+1)} - \vec{x}^{(k)} = -\alpha_k \nabla f\left(\vec{x}^{(k)}\right)$.

Question

alpha k

- So How Should be Choose Learning Rate?

Corollary (Gradient descent is a descent method). *If* $\alpha_k \leq \frac{1}{L}$

$$f\left(\vec{x}^{(k+1)}\right) \leq f\left(\vec{x}^{(k)}\right) - \frac{\alpha_k}{2}\left|\left|\nabla f\left(\vec{x}^{(k)}\right)\right|\right|_2^2.$$

Convergence Analysis

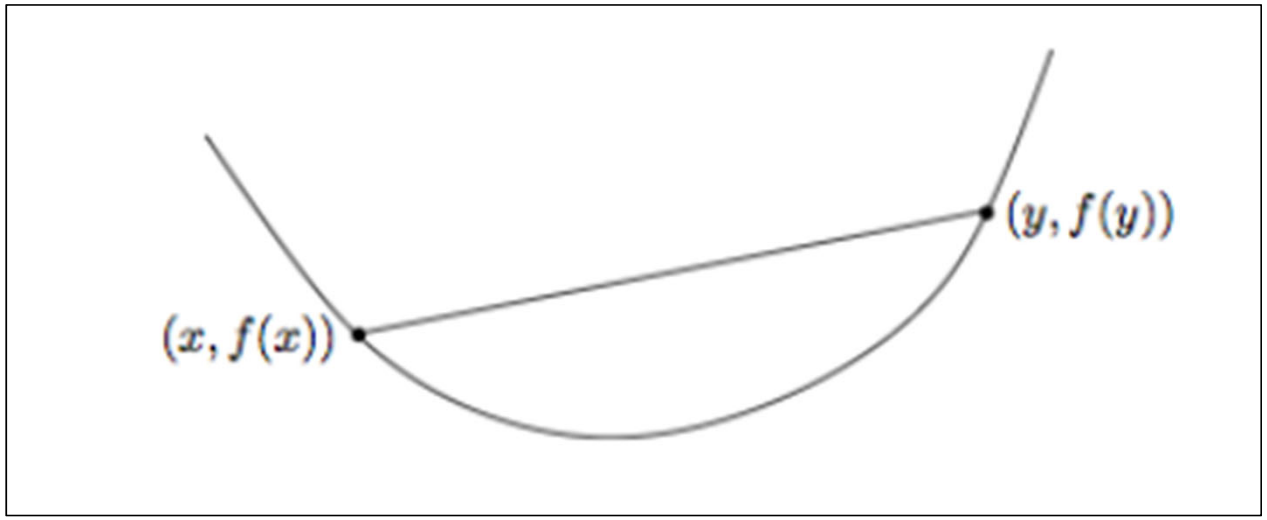
Theorem : *We assume that f is convex, ∇f is L-Lipschitz continuous and there exists a point $\vec{x}^*$ at which f achieves a finite minimum. If we set the step size of gradient descent to $\alpha_k = \alpha \leq 1/L$ for every iteration,*

$$f\left(\vec{x}^{(k)}\right) - f\left(\vec{x}^*\right) \leq \frac{\left|\left|\vec{x}^{(0)} - \vec{x}^*\right|\right|_2^2}{2\,\alpha\,k}$$

non negative

Convex Functions

Definition $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y).$

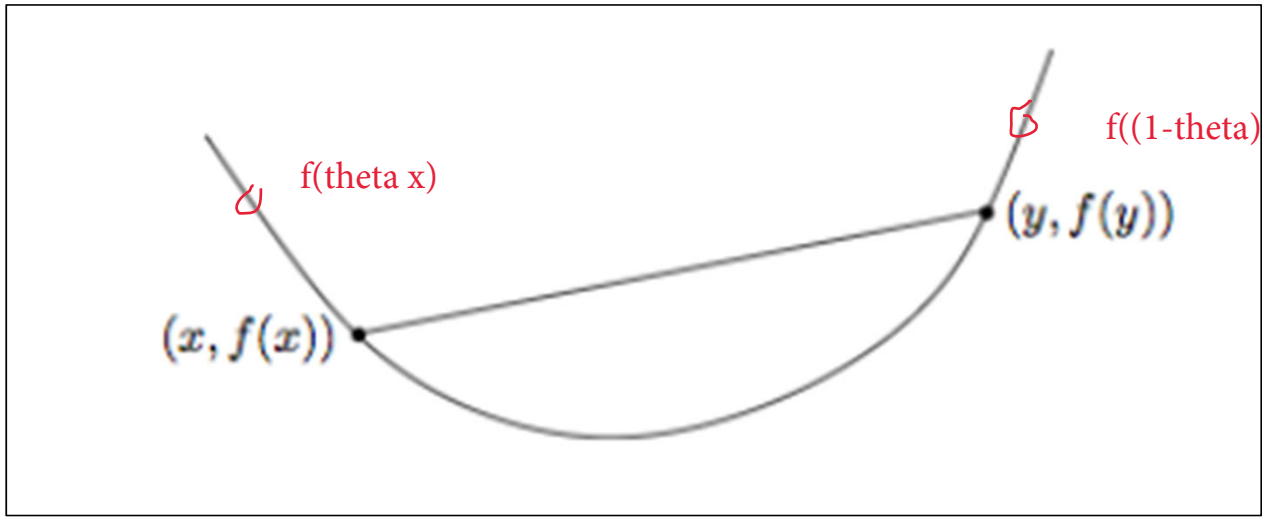


convex: if there is a local minimum, then it is a **global** minimum

Convex Functions

theta from [0,1]

Definition $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y).$

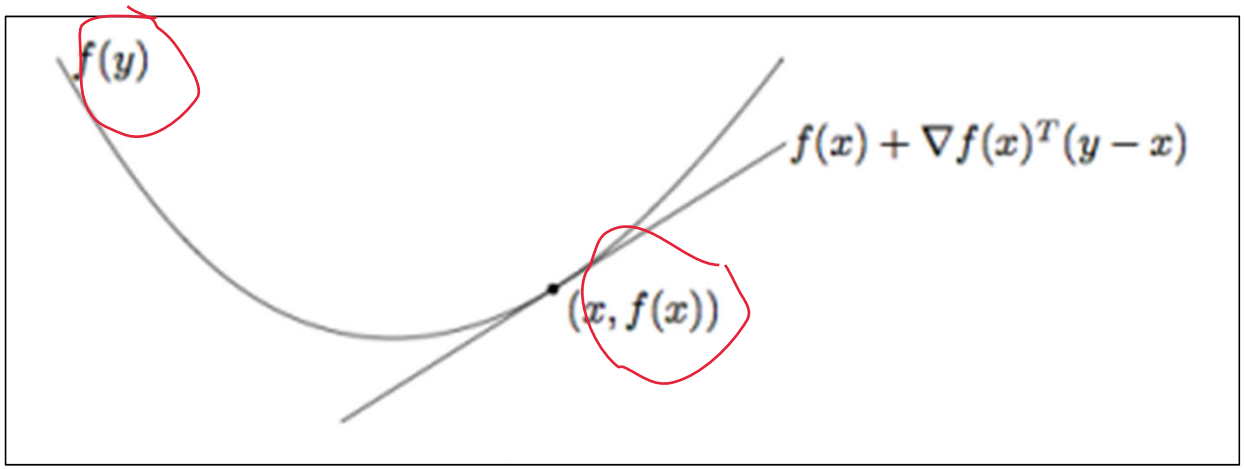


convex: if there is a local minimum, then it is a **global** minimum

Convex Functions

f is convex if and only if

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$



Convergence Analysis

Theorem : *We assume that f is convex, ∇f is L-Lipschitz continuous and there exists a point $\vec{x}^*$ at which f achieves a finite minimum. If we set the step size of gradient descent to $\alpha_k = \alpha \leq 1/L$ for every iteration,*

$$f\left(\vec{x}^{(k)}\right) - f\left(\vec{x}^*\right) \leq \frac{\left|\left|\vec{x}^{(0)} - \vec{x}^*\right|\right|_2^2}{2\,\alpha\,k}$$

Proof. By the first-order characterization of convexity

$$f\left(\vec{x}^{(k-1)}\right) + \nabla f\left(\vec{x}^{(k-1)}\right)^T \left(\vec{x}^* - \vec{x}^{(k-1)}\right) \leq f\left(\vec{x}^*\right),$$

Convergence Analysis

Proof. By the first-order characterization of convexity

$$f\left(\vec{x}^{(k-1)}\right)+\nabla f\left(\vec{x}^{(k-1)}\right)^{T}\left(\vec{x}^{*}-\vec{x}^{(k-1)}\right)\leq f\left(\vec{x}^{*}\right),$$

Gradient descent is a descent method

$$f\left(\vec{x}^{(k+1)}\right)\leq f\left(\vec{x}^{(k)}\right)-\frac{\alpha_k}{2}\left|\left|\nabla f\left(\vec{x}^{(k)}\right)\right|\right|_2^2.$$

$$\begin{aligned}
f\left(\vec{x}^{(k)}\right)-f\left(\vec{x}^{*}\right) &\leq \nabla f\left(\vec{x}^{(k-1)}\right)^{T}\left(\vec{x}^{(k-1)}-\vec{x}^{*}\right)-\frac{\alpha}{2}\left|\left|\nabla f\left(\vec{x}^{(k-1)}\right)\right|\right|_2^2 \\
&= \frac{1}{2\,\alpha}\left(\left|\left|\vec{x}^{(k-1)}-\vec{x}^{*}\right|\right|_2^2-\left|\left|\vec{x}^{(k-1)}-\vec{x}^{*}-\alpha\nabla f\left(\vec{x}^{(k-1)}\right)\right|\right|_2^2\right) \\
&= \frac{1}{2\,\alpha}\left(\left|\left|\vec{x}^{(k-1)}-\vec{x}^{*}\right|\right|_2^2-\left|\left|\vec{x}^{(k)}-\vec{x}^{*}\right|\right|_2\right)
\end{aligned}$$

Convergence Analysis

Theorem : *We assume that f is convex, ∇f is L-Lipschitz continuous and there exists a point $\vec{x}^*$ at which f achieves a finite minimum. If we set the step size of gradient descent to $\alpha_k = \alpha \leq 1/L$ for every iteration,*

$$f\left(\vec{x}^{(k)}\right) - f\left(\vec{x}^*\right) \leq \frac{\left|\left|\vec{x}^{(0)} - \vec{x}^*\right|\right|_2^2}{2\,\alpha\,k}$$

Proof.

$$\begin{aligned} f\left(\vec{x}^{(k)}\right) - f\left(\vec{x}^*\right) &\leq \frac{1}{k}\sum_{i=1}^{k} f\left(\vec{x}^{(k)}\right) - f\left(\vec{x}^*\right) \\ &\leq \frac{1}{2\,\alpha\,k}\left(\left|\left|\vec{x}^{(0)} - \vec{x}^*\right|\right|_2^2 - \left|\left|\vec{x}^{(k)} - \vec{x}^*\right|\right|_2^2\right) \\ &\leq \frac{\left|\left|\vec{x}^{(0)} - \vec{x}^*\right|\right|_2^2}{2\,\alpha\,k}. \end{aligned}$$

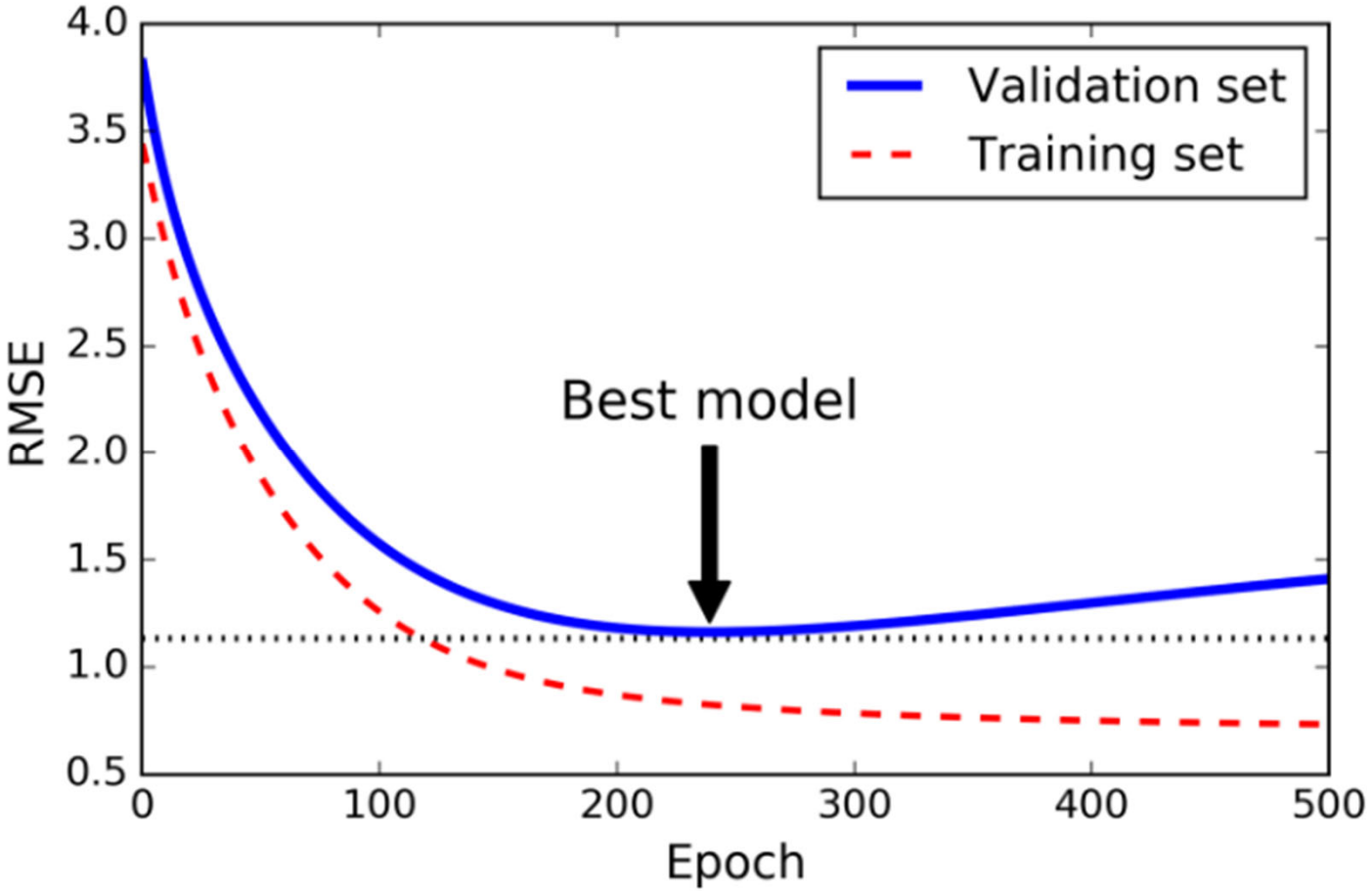
Agenda

- Lesson
  - Properties Ridge
  - Properties of Lasso
  - Relationship between Ridge and Lasso
  - Line Search

Objectives

- Why L2 shrink weights?
- Why L1 gives sparsity?
- What happens with repeated or correlated features.
- Explain line search algorithm

Early Stopping



F_0 in F_1 in F_2

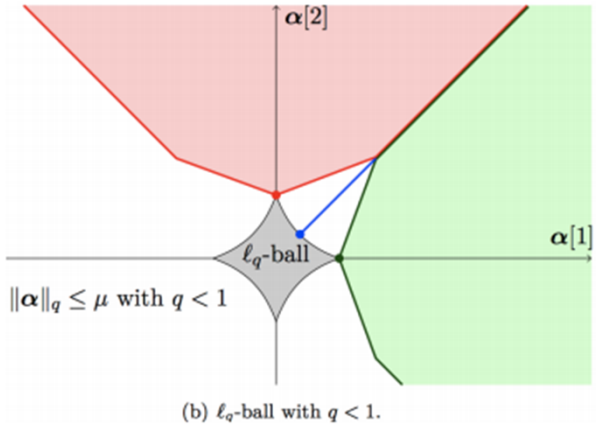
Ridge Regression (Penalization and Constraint Forms)

Ridge Regression (Tikhonov Form)

The ridge r…r $\lambda \geqslant 0$ is

$$\}^2 + \lambda \|w\|_2^2,$$

where $\|w\|$ … n.



(b) ℓ_q-ball with $q < 1$.

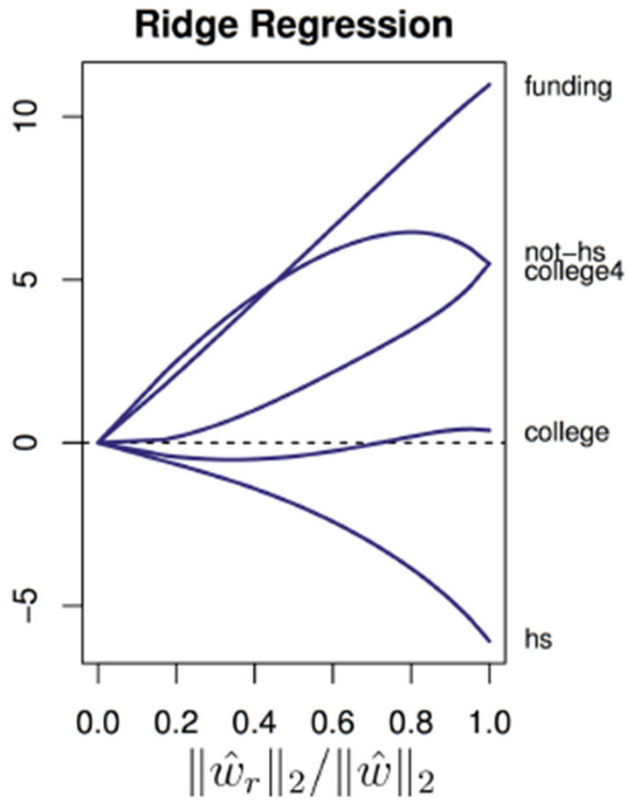
Ridge Regression (Ivanov Form)

The ridge regression solution for complexity parameter $r \geqslant 0$ is

$$\hat{w} = \underset{\|w\|_2^2 \leqslant r^2}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \{w^T x_i - y_i\}^2.$$

why lambda before the l2 norm not before the 1/n,
is that different approach?
yes, only contral the shrink w,
accuracy of the first, other control the complexity of the regression.
lambda small focus on empirical risk which is before

Regularization Path of Ridge Regression



$$\hat{w}_r = \arg\min_{\|w\|_2^2 \le r^2} \frac{1}{n} \sum_{i=1}^{n} \left(w^T x_i - y_i\right)^2$$

$$\hat{w} = \hat{w}_\infty = \text{Unconstrained ERM}$$

- For $r = 0$, $\|\hat{w}_r\|_2/\|\hat{w}\|_2 = 0$.
- For $r = \infty$, $\|\hat{w}_r\|_2/\|\hat{w}\|_2 = 1$

the weight of regularization/ non regular

always between [0,1], uncontrain ,weight should big than contrained(only to small)

Ridge Regression and Stability

$$\hat{f}(x) = \hat{w}^T x,$$

$$\begin{aligned} \left|\hat{f}(x+h) - \hat{f}(x)\right| &= |\hat{w}^T(x+h) - \hat{w}^T x| = \left|\hat{w}^T h\right| \\ &\leqslant \|\hat{w}\|_2 \|h\|_2 \end{aligned}$$

small change input also small change in output,
<= Lh

Ridge Regression Shrinks Weights

$$\underset{w \in \mathbb{R}^m}{\operatorname{argmin}} \left(\frac{1}{2m} \sum_{i=1}^{m} (x_i w - y_i)^2 + \lambda w^2 \right).$$

Then, the optimal solution is

$$w = \frac{\langle \mathbf{x}, \mathbf{y} \rangle / m}{\|\mathbf{x}\|^2 / m + 2\lambda}.$$

one dimension, one feature one label.
drevi

lambda larger, the weight shrinking
(lambda I + X^TX)-1 X^T y = weight

Ridge Regression and Line Search

- Why can't we just solve for the minimum in line search?
- Actually we can…sort of…it's the Levenberg-Marquardt algorithm
- While it's an iterative algorithm, the important step in the algorithm involves solving the normal equations.

Ridge Regression and Line Search

- Why can't we just solve for the minimum in line search?
- Actually we can…sort of…it's the Levenberg-Marquardt algorithm
- While it's an iterative algorithm, the important step in the algorithm involves solving the normal equations.

$$f(x_i, \boldsymbol{\beta} + \boldsymbol{\delta}) \approx f(x_i, \boldsymbol{\beta}) + \mathbf{J}_i \boldsymbol{\delta},$$

where

$$\mathbf{J}_i = \frac{\partial f(x_i, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$$

Levenberg-Marquardt algorithm

$$S(\boldsymbol{\beta}+\boldsymbol{\delta}) \approx \sum_{i=1}^{m} \left[y_i - f(x_i, \boldsymbol{\beta}) - \mathbf{J}_i \boldsymbol{\delta}\right]^2,$$

$$\begin{aligned}
S(\boldsymbol{\beta}+\boldsymbol{\delta}) &\approx \|\mathbf{y} - \mathbf{f}(\boldsymbol{\beta}) - \mathbf{J}\boldsymbol{\delta}\|^2 \\
&= [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta}) - \mathbf{J}\boldsymbol{\delta}]^{\mathrm{T}} [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta}) - \mathbf{J}\boldsymbol{\delta}] \\
&= [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^{\mathrm{T}} [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})] - [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^{\mathrm{T}} \mathbf{J}\boldsymbol{\delta} - (\mathbf{J}\boldsymbol{\delta})^{\mathrm{T}} [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})] + \boldsymbol{\delta}^{\mathrm{T}} \mathbf{J}^{\mathrm{T}} \mathbf{J}\boldsymbol{\delta} \\
&= [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^{\mathrm{T}} [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})] - 2[\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^{\mathrm{T}} \mathbf{J}\boldsymbol{\delta} + \boldsymbol{\delta}^{\mathrm{T}} \mathbf{J}^{\mathrm{T}} \mathbf{J}\boldsymbol{\delta}.
\end{aligned}$$

$$\left(\mathbf{J}^{\mathrm{T}}\mathbf{J}\right)\boldsymbol{\delta} = \mathbf{J}^{\mathrm{T}}\left[\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})\right],$$

Levenberg-Marquardt algorithm

$$S(\boldsymbol{\beta}+\boldsymbol{\delta}) \approx \sum_{i=1}^{m} \left[y_i - f(x_i, \boldsymbol{\beta}) - \mathbf{J}_i \boldsymbol{\delta}\right]^2,$$

$$\begin{aligned} S(\boldsymbol{\beta}+\boldsymbol{\delta}) &\approx \|\mathbf{y} - \mathbf{f}(\boldsymbol{\beta}) - \mathbf{J}\boldsymbol{\delta}\|^2 \\ &= [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta}) - \mathbf{J}\boldsymbol{\delta}]^{\mathrm{T}} [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta}) - \mathbf{J}\boldsymbol{\delta}] \\ &= [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^{\mathrm{T}} [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})] - [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^{\mathrm{T}} \mathbf{J}\boldsymbol{\delta} - (\mathbf{J}\boldsymbol{\delta})^{\mathrm{T}} [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})] + \boldsymbol{\delta}^{\mathrm{T}} \mathbf{J}^{\mathrm{T}} \mathbf{J}\boldsymbol{\delta} \\ &= [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^{\mathrm{T}} [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})] - 2[\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^{\mathrm{T}} \mathbf{J}\boldsymbol{\delta} + \boldsymbol{\delta}^{\mathrm{T}} \mathbf{J}^{\mathrm{T}} \mathbf{J}\boldsymbol{\delta}. \end{aligned}$$

$$\boxed{\left(\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda \mathbf{I}\right) \boldsymbol{\delta} = \mathbf{J}^{\mathrm{T}} [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]}$$

Lasso Regression (Constraint and Penalization Forms)

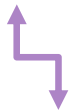
Lasso Regression (Tikhonov Form)

The lasso regression solution for regularization parameter $\lambda \geqslant 0$ is

$$\hat{w} = \arg\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left\{ w^T x_i - y_i \right\}^2 + \lambda \|w\|_1,$$

where $\|w\|_1 = |w_1| + \cdots + |w_d|$ is the ℓ_1-norm.

ll1 don't shrink the weight but sparisity

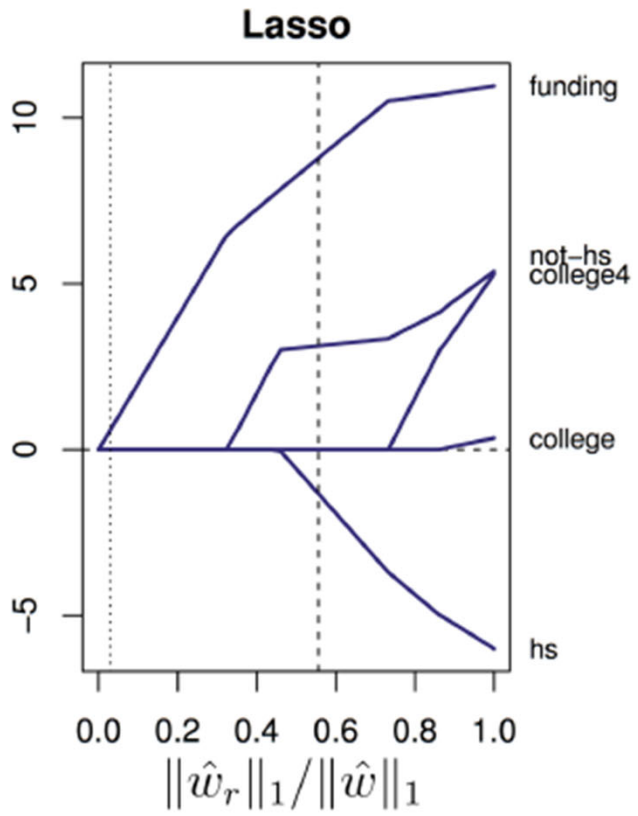


Lasso Regression (Ivanov Form)

The lasso regression solution for complexity parameter $r \geqslant 0$ is

$$\hat{w} = \arg\min_{\|w\|_1 \leqslant r} \frac{1}{n} \sum_{i=1}^{n} \left\{ w^T x_i - y_i \right\}^2.$$

Regularization Path of Lasso Regression



$$\hat{w}_r = \arg\min_{\|w\|_1 \le r} \frac{1}{n} \sum_{i=1}^{n} \left(w^T x_i - y_i\right)^2$$

$$\hat{w} = \hat{w}_\infty = \text{Unconstrained ERM}$$

- For $r = 0$, $\|\hat{w}_r\|_1/\|\hat{w}\|_1 = 0$.
- For $r = \infty$, $\|\hat{w}_r\|_1/\|\hat{w}\|_1 = 1$

four features,weight is 0 means sparse

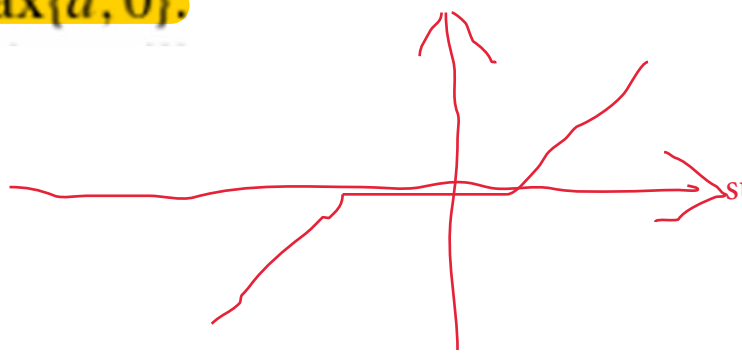
Lasso Regression Gives Sparsity

$$\min_{w \in \mathbb{R}} \left(\frac{1}{2} w^2 - x w + \lambda |w| \right).$$

solution to this problem is the "soft thresholding" operator

$$w = \text{sign}(x) \left[|x| - \lambda \right]_+ ,$$

where $[a]_+ \stackrel{\text{def}}{=} \max\{a, 0\}$.



w is positive: w - x + lambda

= 0

is w = x-lambda > =0

x >= lambda

subgradient:

Lasso Regression Gives Sparsity

$$\operatorname*{argmin}_{w \in \mathbb{R}^m} \left(\frac{1}{2m} \sum_{i=1}^{m} (x_i w - y_i)^2 + \lambda |w| \right).$$

We can rewrite the problem as

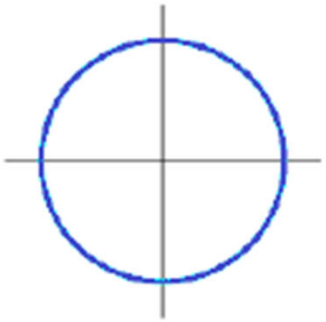
$$\operatorname*{argmin}_{w \in \mathbb{R}^m} \left(\frac{1}{2} \left(\frac{1}{m} \sum_i x_i^2 \right) w^2 - \left(\frac{1}{m} \sum_{i=1}^{m} x_i y_i \right) w + \lambda |w| \right).$$

For simplicity let us assume that $\frac{1}{m} \sum_i x_i^2 = 1$, and denote $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{m} x_i y_i$; then the optimal solution is

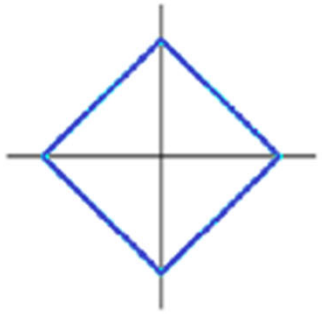
$$w = \operatorname{sign}(\langle \mathbf{x}, \mathbf{y} \rangle) \left[|\langle \mathbf{x}, \mathbf{y} \rangle| / m - \lambda \right]_+ .$$

Lasso Regression Gives Sparsity

- ℓ_2 contour:
  $w_1^2 + w_2^2 = r$

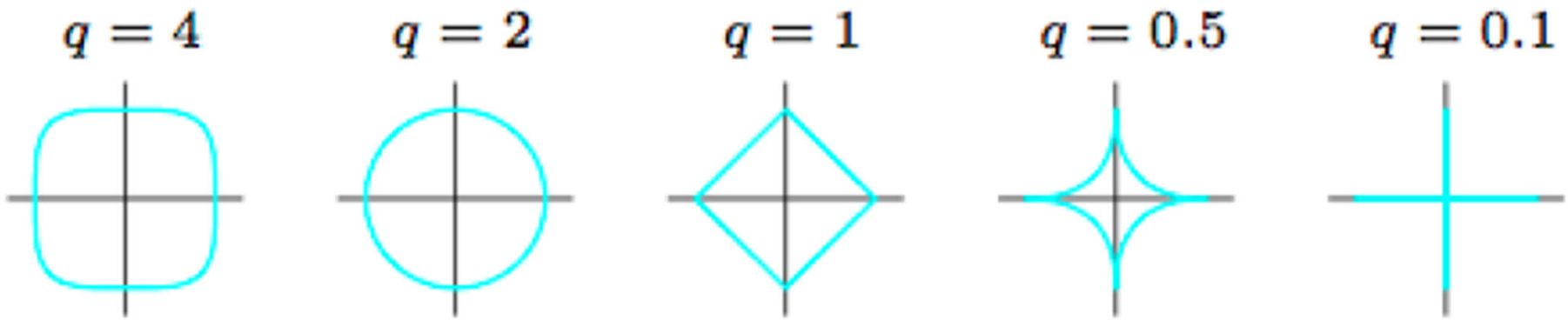


- ℓ_1 contour:
  $|w_1| + |w_2| = r$

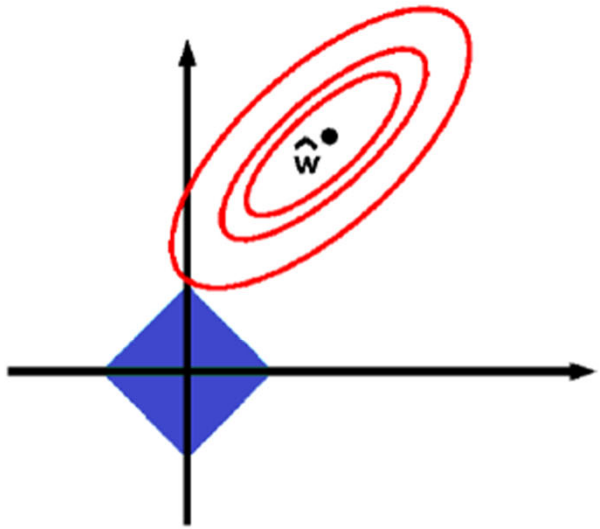


Lasso Regression Gives Sparsity

manhatan dist



Lasso Regression Gives Sparsity



- Blue region: Area satisfying complexity constraint: $|w_1| + |w_2| \leqslant r$
- Red lines: contours of $\hat{R}_n(w) = \sum_{i=1}^{n} \left(w^T x_i - y_i\right)^2$.

Lasso Regression Gives Sparsity

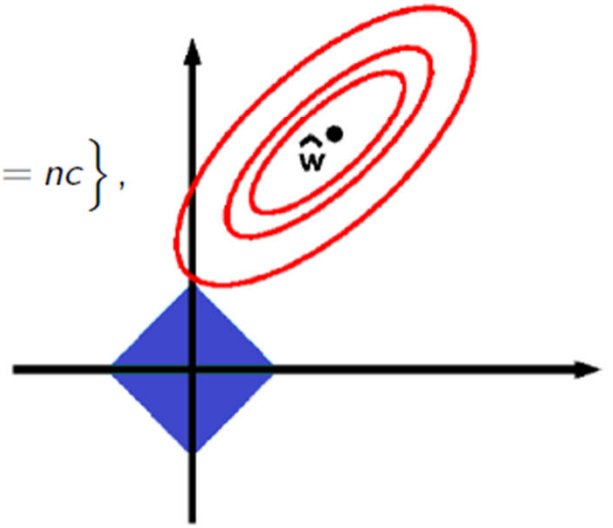
- By "completing the square", we can show for any $w \in \mathbf{R}^d$:

$$\hat{R}_n(w) = \frac{1}{n}(w-\hat{w})^T X^T X (w-\hat{w}) + \hat{R}_n(\hat{w})$$

- Set of w with $\hat{R}_n(w)$ exceeding $\hat{R}_n(\hat{w})$ by $c > 0$ is

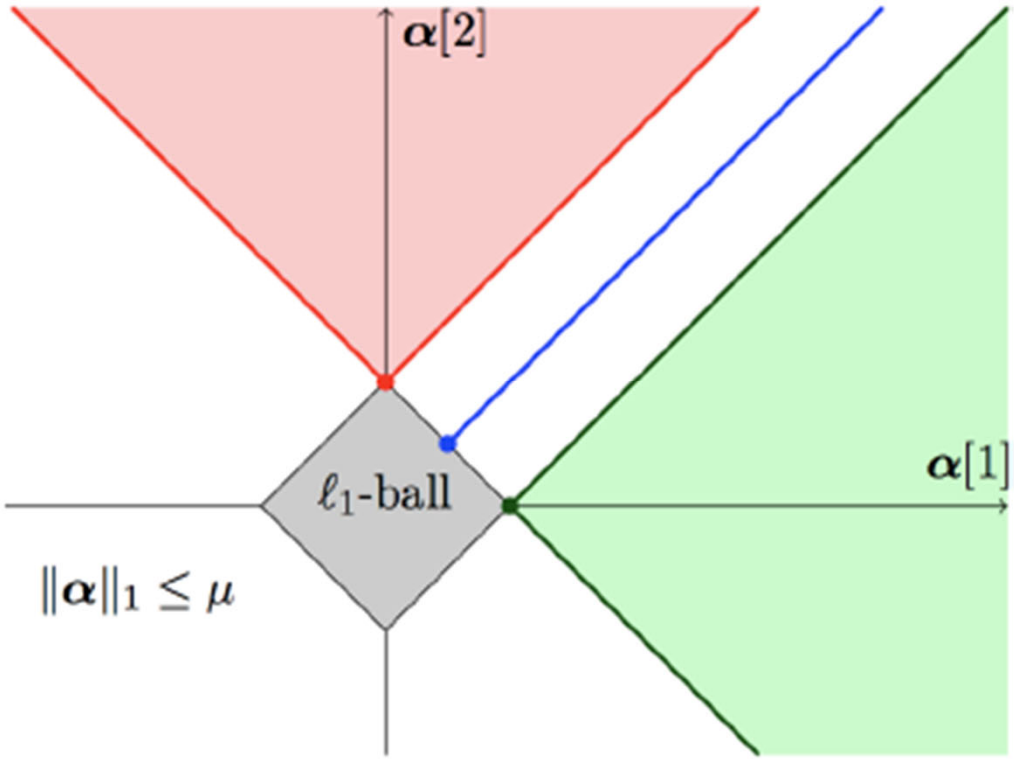
$$\left\{w \mid \hat{R}_n(w) = c + \hat{R}_n(\hat{w})\right\} = \left\{w \mid (w-\hat{w})^T X^T X (w-\hat{w}) = nc\right\},$$

which is an **ellipsoid centered at** $\hat{w}$.

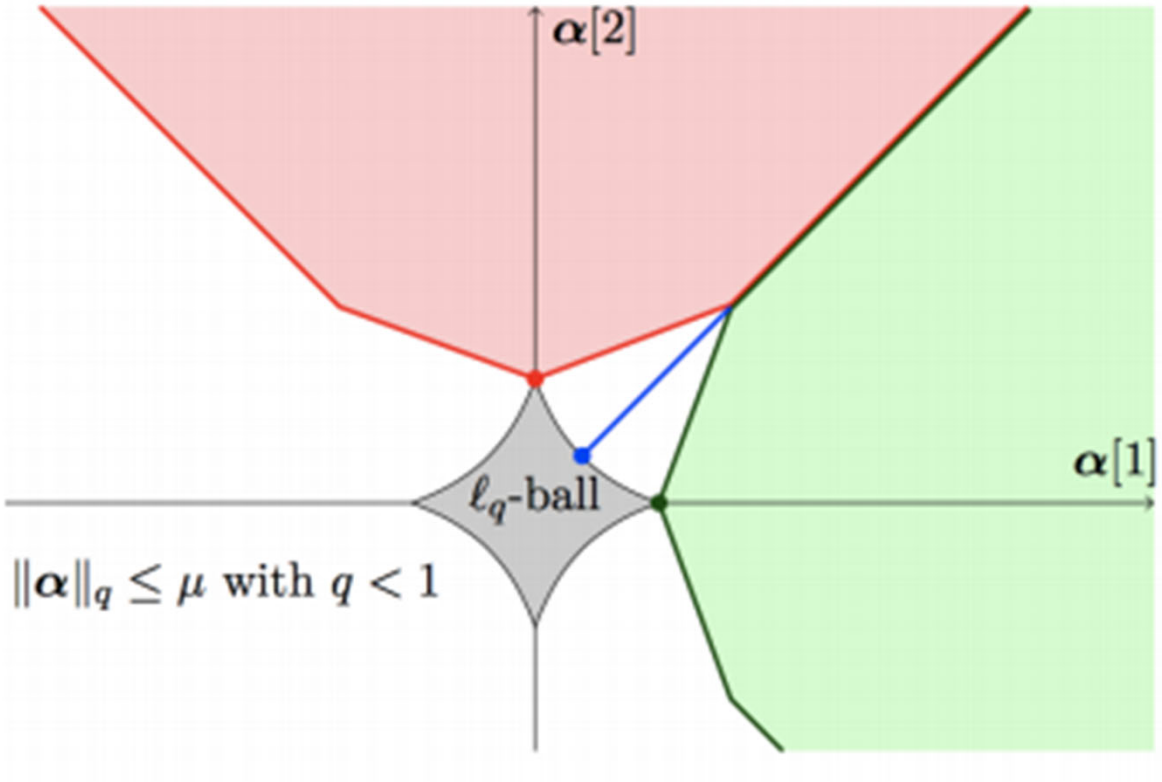


- Blue region: Area satisfying complexity constraint: $|w_1| + |w_2| \leqslant r$
- Red lines: contours of $\hat{R}_n(w) = \sum_{i=1}^n \left(w^T x_i - y_i\right)^2$.

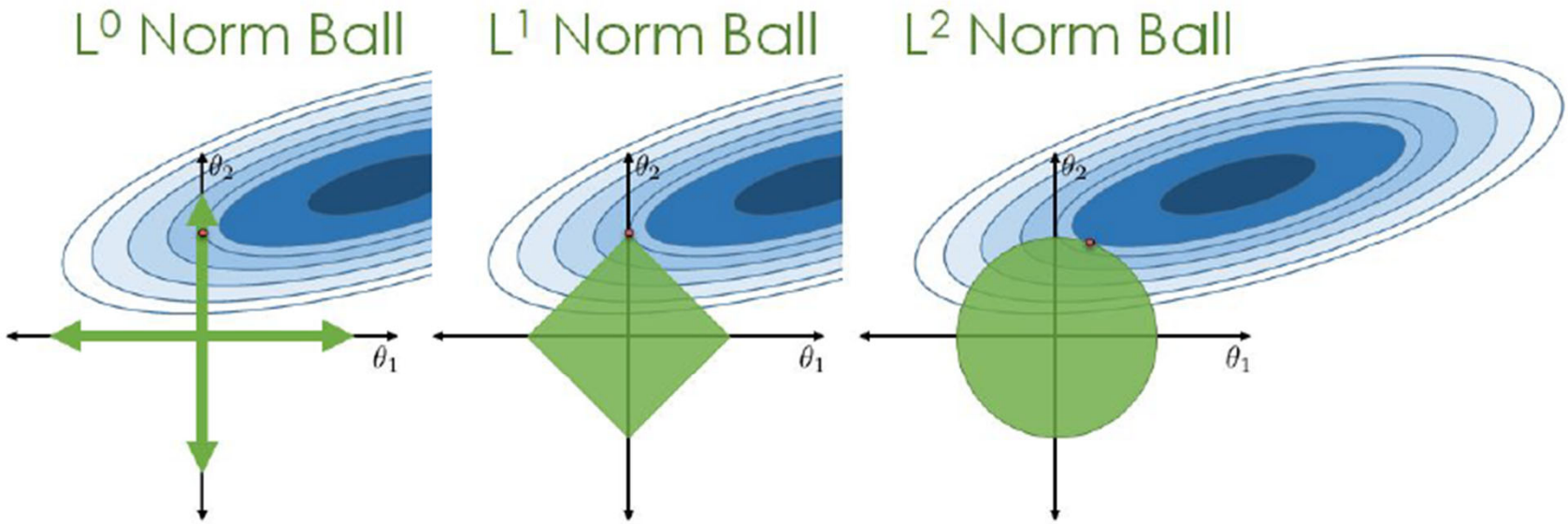
Lasso Regression Gives Sparsity



Lasso Regression Gives Sparsity



Lasso Regression Gives Sparsity



Ideal for
Feature Selection
but combinatorically difficult to optimize

Encourages
Sparse Solutions
Convex!

Spreads weight
over features **(robust)**
does not
encourage sparsity

Lasso Regression Gives Sparsity

- Coefficient are 0 = > don't need those features. What's the gain?
  - Time/expense to compute/buy features
  - Memory to store features (e.g. real-time deployment)
  - Identifies the important features
  - Better prediction? sometimes
  - As a feature-selection step for training a slower non-linear model

Question

- Explain why feature normalization is important if you are using L1 or L2 regularization.
- Scaling
  - Centering and Scaling
  - Clipping
  - Min-Max
  - Transformation (Log, Sigmoid,…)

How to work with absolute value?

- How to solve the Lasso?

$$\min_{w \in \mathbf{R}^d} \sum_{i=1}^{n} \left(w^T x_i - y_i\right)^2 + \lambda \|w\|_1$$

- $\|w\|_1 = |w_1| + |w_2|$ is not differentiable!

How to work with absolute value?

- Consider any number $a \in \mathbf{R}$.
- Let the **positive part** of a be
$$a^+ = a1(a \geqslant 0).$$
- Let the **negative part** of a be
$$a^- = -a1(a \leqslant 0).$$
- Do you see why $a^+ \geqslant 0$ and $a^- \geqslant 0$?
- How do you write a in terms of a^+ and a^-?
- How do you write $|a|$ in terms of a^+ and a^-?

How to work with absolute value?

- The Lasso problem

$$\min_{w \in \mathbf{R}^d} \sum_{i=1}^{n} \left(w^T x_i - y_i \right)^2 + \lambda \|w\|_1$$

- Replace each w_i by $w_i^+ - w_i^-$.
- Write $w^+ = \left(w_1^+, \ldots, w_d^+ \right)$ and $w^- = \left(w_1^-, \ldots, w_d^- \right)$.

How to work with absolute value?

We **will show:** substituting $w = w^+ - w^-$ and $|w| = w^+ + w^-$ gives an **equivalent** problem:

$$\begin{aligned} \min_{w^+, w^-} \quad & \sum_{i=1}^{n} \left(\left(w^+ - w^-\right)^T x_i - y_i \right)^2 + \lambda 1^T \left(w^+ + w^-\right) \\ \text{subject to} \quad & w_i^+ \geqslant 0 \text{ for all } i \qquad w_i^- \geqslant 0 \text{ for all } i, \end{aligned}$$

- Objective is **differentiable** (in fact, **convex and quadratic**)
- $2d$ variables vs d variables and $2d$ constraints vs no constraints

How to work with absolute value?

Lasso problem is trivially equivalent to the following:

$$\begin{aligned} \min_{w} \min_{a,b} \quad & \sum_{i=1}^{n} \left((a-b)^T x_i - y_i \right)^2 + \lambda 1^T (a+b) \\ \text{subject to} \quad & a_i \geqslant 0 \text{ for all } i \qquad b_i \geqslant 0 \text{ for all } i, \\ & a - b = w \\ & a + b = |w| \end{aligned}$$

- Claim: Don't need constraint $a+b=|w|$.
- $a' \leftarrow a - \min(a,b)$ and $b' \leftarrow b - \min(a,b)$ at least as good
- So if a and b are minimizers, at least one is 0.
- Since $a-b=w$, we must have $a=w^+$ and $b=w^-$. So also $a+b=|w|$.

How to work with absolute value?

- So lasso optimization problem is equivalent to

$$\begin{aligned} \min_{a,b} \quad & \sum_{i=1}^{n} \left((a-b)^T x_i - y_i \right)^2 + \lambda 1^T (a+b) \\ \text{subject to} \quad & a_i \geqslant 0 \text{ for all } i \qquad b_i \geqslant 0 \text{ for all } i, \end{aligned}$$

  where at the end we take $w^* = a^* - b^*$ (and we've shown above that a^* and b^* are positive and negative parts of w^*, respectively.)

- Has constraints – how do we optimize?

Projected (Stochastic) Gradient Descent

$$\min_{w^+, w^- \in \mathbf{R}^d} \sum_{i=1}^{n} \left(\left(w^+ - w^- \right)^T x_i - y_i \right)^2 + \lambda 1^T \left(w^+ + w^- \right)$$

$$\text{subject to } w_i^+ \geqslant 0 \text{ for all } i$$

$$w_i^- \geqslant 0 \text{ for all } i$$

- Just like SGD, but after each step
  - Project w^+ and w^- into the constraint set.
  - In other words, if any component of w^+ or w^- becomes negative, set it back to 0.

Lasso Regression and Line Search

- **Goal:** Minimize $L(w) = L(w_1, \ldots, w_d)$ over $w = (w_1, \ldots, w_d) \in \mathbf{R}^d$.
- In gradient descent or SGD,
  - each step potentially changes all entries of w.
- In each step of **coordinate descent**,
  - we adjust only a single w_i.
- In each step, solve

$$w_i^{\text{new}} = \arg\min_{w_i} L(w_1, \ldots, w_{i-1}, \mathbf{w_i}, w_{i+1}, \ldots, w_d)$$

- Solving this argmin may itself be an iterative process.
- Coordinate descent is great when
  - it's easy or easier to minimize w.r.t. one coordinate at a time

Shooting Algorithm

$$\hat{w}_j = \arg\min_{w_j \in \mathbf{R}} \sum_{i=1}^{n} \left(w^T x_i - y_i\right)^2 + \lambda |w|_1$$

Then

$$\hat{w}_j = \begin{cases} (c_j + \lambda)/a_j & \text{if } c_j < -\lambda \\ 0 & \text{if } c_j \in [-\lambda, \lambda] \\ (c_j - \lambda)/a_j & \text{if } c_j > \lambda \end{cases}$$

$$a_j = 2\sum_{i=1}^{n} x_{i,j}^2 \qquad\qquad c_j = 2\sum_{i=1}^{n} x_{i,j}(y_i - w_{-j}^T x_{i,-j})$$

where w_{-j} is w without component j and similarly for $x_{i,-j}$.

Shooting Algorithm

- Lasso
  - Explain what coordinate descent is, and why it is of particular interest for the Lasso.
  - Lasso optimization problem does not have a differentiable objective function. Give an equivalent formulation that has a differentiable objective function by dividing the weight vector into positive and negative parts.
  - Give reasons why we might want the sparsity that L1 regularization often provides.

Lasso Regression and Stability

- Suppose we have one feature $x_1 \in \mathbf{R}$.
- Response variable $y \in \mathbf{R}$.
- Got some data and ran least squares linear regression.
- The ERM is

$$\hat{f}(x_1) = 4x_1.$$

- What happens if we get a new feature x_2,
  - but we always have $x_2 = x_1$?

Lasso Regression and Stability

- New feature x_2 gives no new information.
- ERM is still

$$\hat{f}(x_1, x_2) = 4x_1.$$

- Now there are some more ERMs:

$$\begin{aligned} \hat{f}(x_1, x_2) &= 2x_1 + 2x_2 \\ \hat{f}(x_1, x_2) &= x_1 + 3x_2 \\ \hat{f}(x_1, x_2) &= 4x_2 \end{aligned}$$

- What if we introduce ℓ_1 or ℓ_2 regularization?

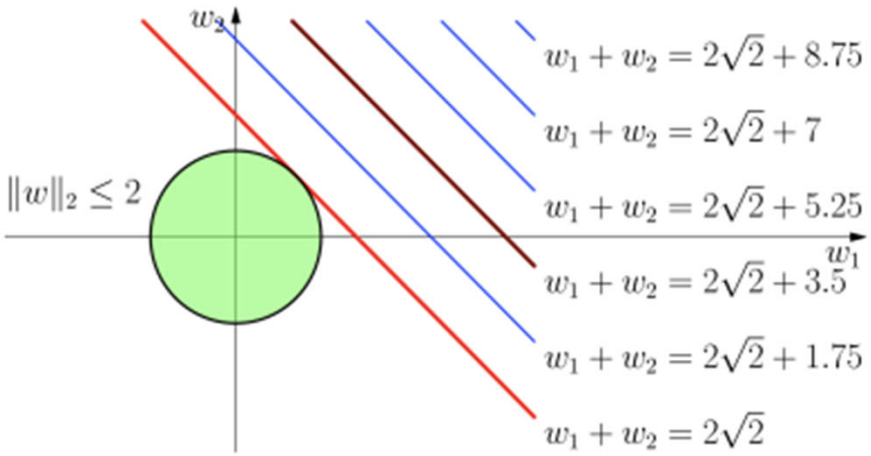
Lasso Regression and Stability

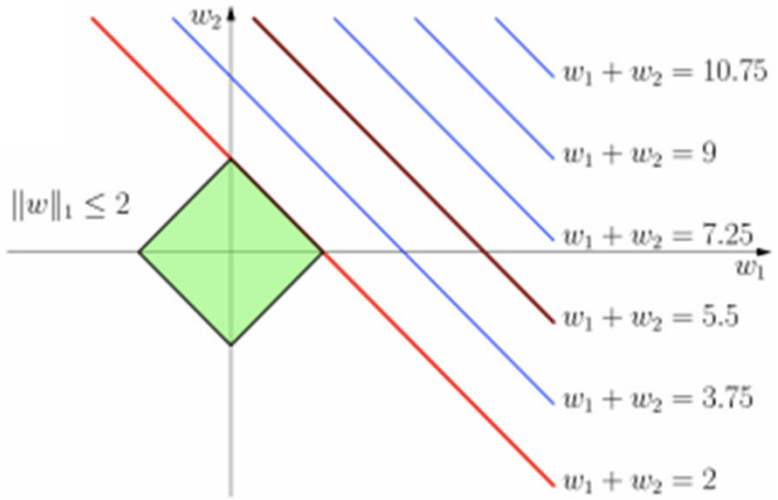
- $\hat{f}(x_1, x_2) = w_1 x_1 + w_2 x_2$ is an ERM iff $w_1 + w_2 = 4$.
- Consider the ℓ_1 and ℓ_2 norms of various solutions:

w_1	w_2	$\|w\|_1$	$\|w\|_2^2$
4	0	**4**	16
2	2	**4**	**8**
1	3	**4**	10
-1	5	6	26

- $\|w\|_1$ doesn't discriminate, as long as all have same sign
- $\|w\|_2^2$ minimized when weight is spread equally
- Picture proof: Level sets of loss are lines of the form $w_1 + w_2 = 4$...

Lasso Regression and Stability





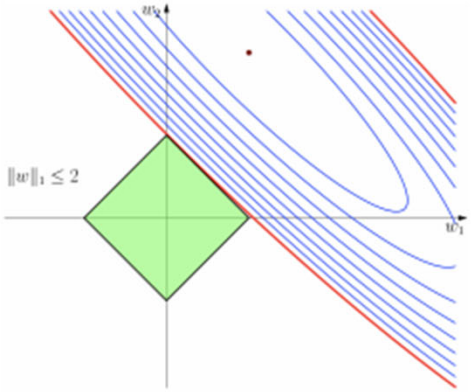
Lasso Regression and Stability

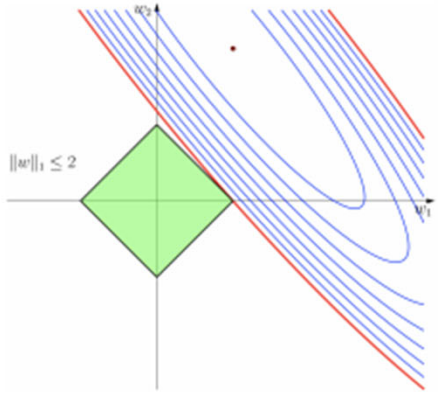
- Suppose x_1 and x_2 are highly correlated and the same scale.
- This is quite typical in real data, after normalizing data.

- Nothing degenerate here, so level sets are ellipsoids.

- But, the higher the correlation, the closer to degenerate we get.
- That is, ellipsoids keep stretching out, getting closer to two parallel lines.

Lasso Regression and Stability





- Intersection could be anywhere on the top right edge.
- Minor perturbations (in data) can drastically change intersection point – very unstable solution.
- Makes division of weight among highly correlated features (of same scale) seem arbitrary.
  - If $x_1 \approx 2x_2$, ellipse changes orientation and we hit a corner. (Which one?)

Lasso Regression and Stability

- Suppose there's some unknown value $\theta \in \mathbf{R}$.
- We get 3 noisy observations of θ:

$$x_1, x_2, x_3 \sim \mathcal{N}(\theta, 1) \text{ (i.i.d)}$$

- What's a good estimator $\hat{\theta}$ for θ?
- Would you prefer $\hat{\theta} = x_1$ or $\hat{\theta} = \frac{1}{3}(x_1 + x_2 + x_3)$?

Lasso Regression and Stability

- $\mathbb{E}[x_1] = \theta$ and $\mathbb{E}\left[\frac{1}{3}(x_1 + x_2 + x_3)\right] = \theta$. So both unbiased.
- $\mathrm{Var}[x_1] = 1$.
- $\mathrm{Var}\left[\frac{1}{3}(x_1 + x_2 + x_3)\right] = \frac{1}{9}(1+1+1) = \frac{1}{3}$.
- Average has a smaller variance — the independent errors cancel each other out.
- Similar thing happens in regression with correlated features:
  - e.g. If 3 features are correlated, we could keep just one of them.
  - But we can potentially do better by using all 3.

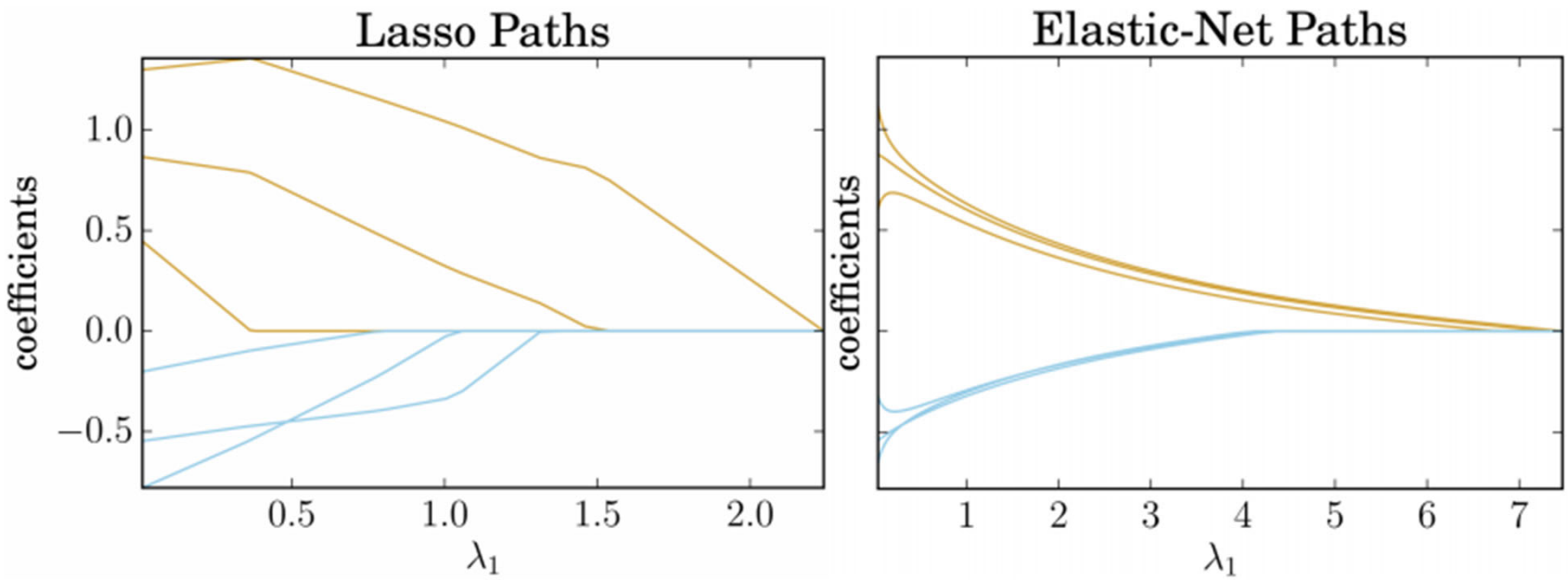
Elastic Net

- The **elastic net** combines lasso and ridge penalties:

$$\hat{w} = \arg\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left\{ w^T x_i - y_i \right\}^2 + \lambda_1 \|w\|_1 + \lambda_2 \|w\|_2^2$$

- We expect correlated random variables to have similar coefficients.

Elastic Net


Lasso Paths
coefficients
1.0
0.5
0.0
−0.5
0.5
1.0
1.5
2.0
λ_1
Elastic-Net Paths
coefficients
1
2
3
4
5
6
7
λ_1

Take-Aways

- Lasso
  - Explain what coordinate descent is, and why it is of particular interest for the Lasso.
  - Lasso optimization problem does not have a differentiable objective function. Give an equivalent formulation that has a differentiable objective function by dividing the weight vector into positive and negative parts.
  - Give reasons why we might want the sparsity that L1 regularization often provides.

Take-Aways

- Ridge versus Lasso
  - Explain what happens when we do linear, lasso, and ridge regression with 2 identical features.
  - Explain what happens when x1 and x2 are highly correlated, but not exactly linearly related.
  - Why does Elastic Net provide us benefits of both L1 and L2 regularization?