NYU Center for Data Science

DS-GA 3001.007
Introduction to Machine Learning

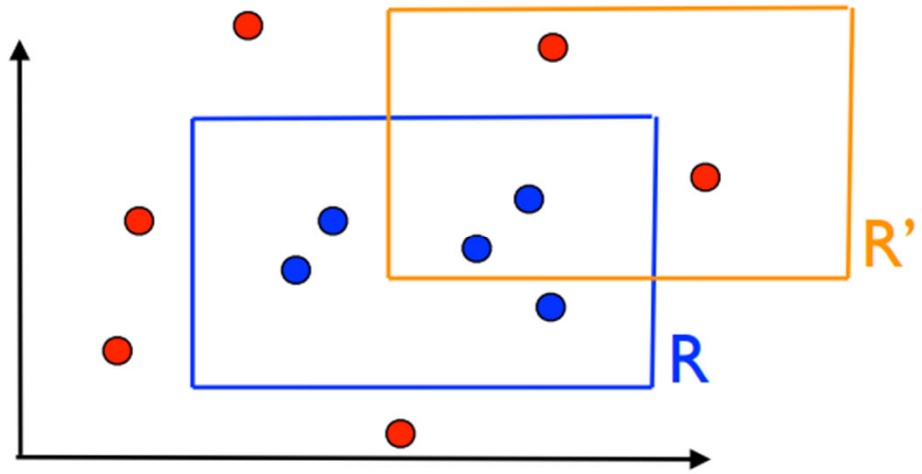
Lecture 5
Regularization – Ridge and Lasso

Reminders

- Homework 3
  - Due on Sunday October 20
- Midterm
  - In class on Wednesday October 23
- Project
  - Proposal due October 31
  - Milestone due November 28
  - Report due December 15

Bound Difference In Sample and Out of Sample

- Problem: learn unknown axis-aligned rectangle R using as small a labeled sample as possible.



complexity of the model
sample comp
model comp
how to avoid the overfitting
$P(x,y) = P(X)*P(y|X)$
two var for x, 2 dimension
four parameter,
minimize the blue box
R' should be in R

- Hypothesis: rectangle R'. In general, there may be false positive and false negative points.

Bound Difference In Sample and Out of Sample

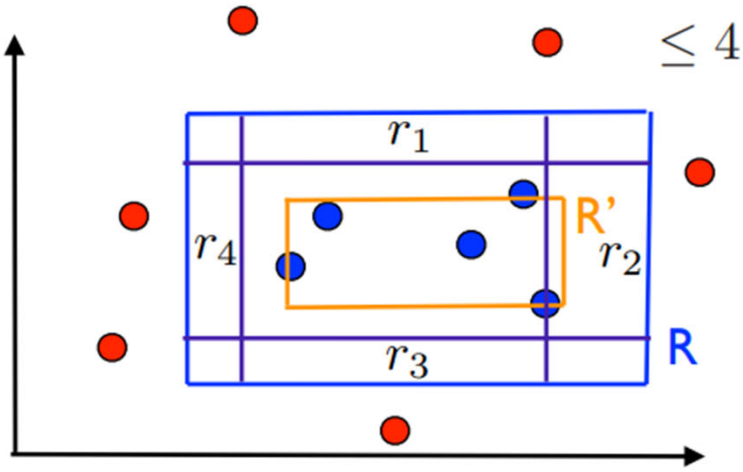
- Errors can only occur in $\mathrm{R}-\mathrm{R}'$. Thus (geometry),

$$R(\mathrm{R}') > \epsilon \Rightarrow \mathrm{R}' \text{ misses at least one region } r_i.$$

- Therefore, $\Pr[R(\mathrm{R}') > \epsilon] \leq \Pr[\cup_{i=1}^{4} \{\mathrm{R}' \text{ misses } r_i\}]$

$$\leq \sum_{i=1}^{4} \Pr[\{\mathrm{R}' \text{ misses } r_i\}]$$

$$\leq 4(1-\tfrac{\epsilon}{4})^m \leq 4e^{-\frac{m\epsilon}{4}}.$$



but miss classification
m is the number of samples

Bound Difference In Sample and Out of Sample

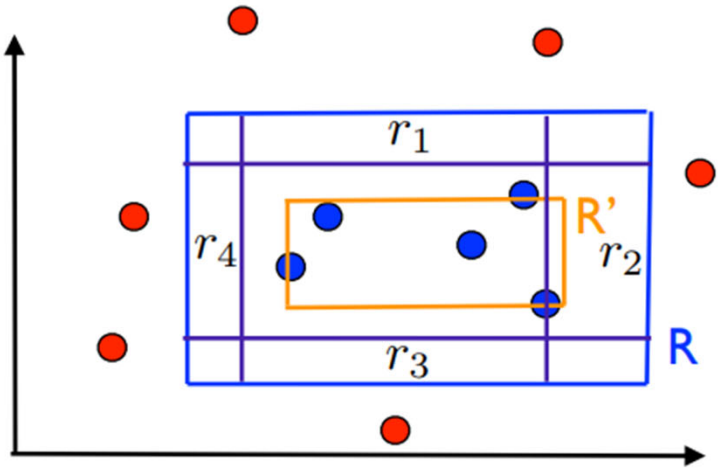
- Set $\delta > 0$ to match the upper bound:

$$4e^{-\frac{m\epsilon}{4}} \leq \delta \Leftrightarrow m \geq \frac{4}{\epsilon} \log \frac{4}{\delta}.$$

epsa can be a half,

- Then, for $m \geq \frac{4}{\epsilon} \log \frac{4}{\delta}$, with probability at least $1-\delta$,

$$R(\mathrm{R}') \leq \epsilon.$$

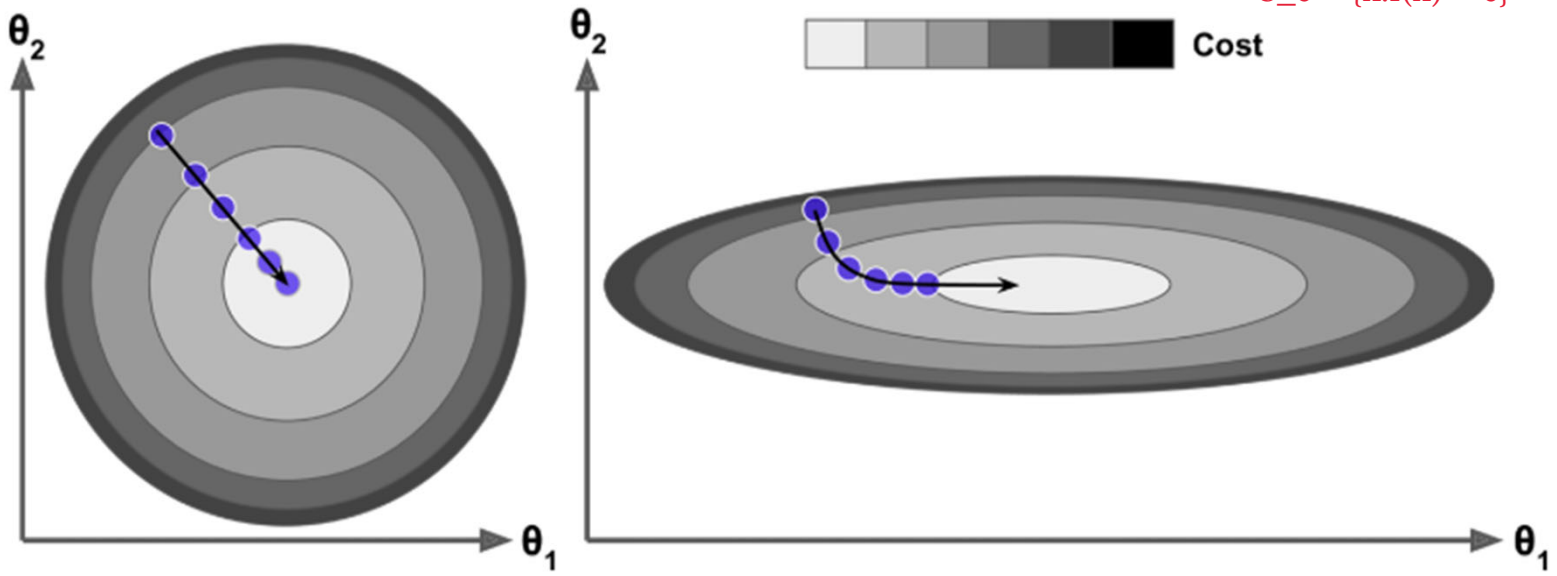


Gradient Descent

apptimaize

two features and two weights
square loss with GD

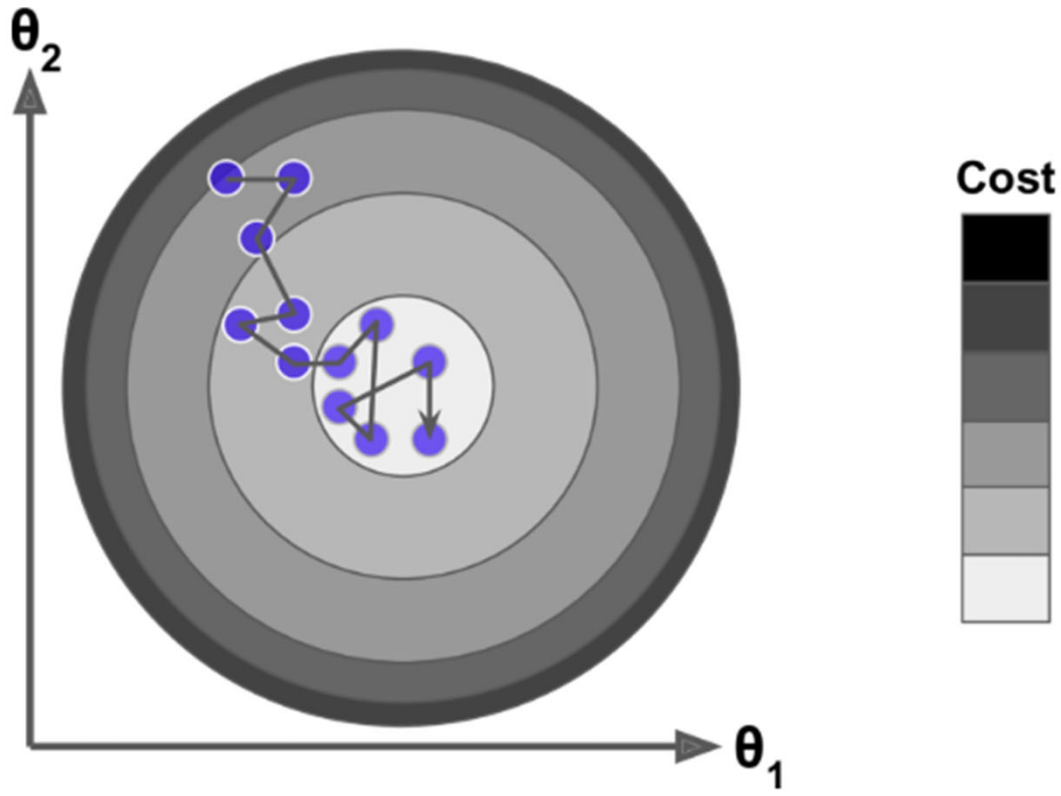
level set: function f -> R^2 to R
S_c = {x:f(x) = c}



Gradient Descent

- Initialize $x = 0$
- repeat
  - $x \leftarrow x - \underbrace{\eta}_{\text{step size}} \nabla f(x)$
- until stopping criterion satisfied

Stochastic Gradient Descent

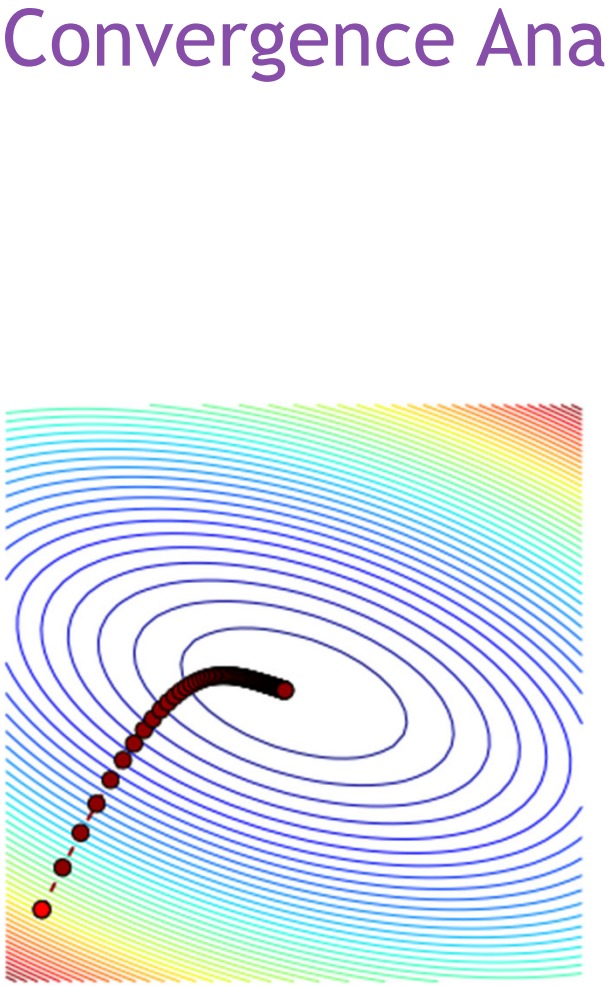


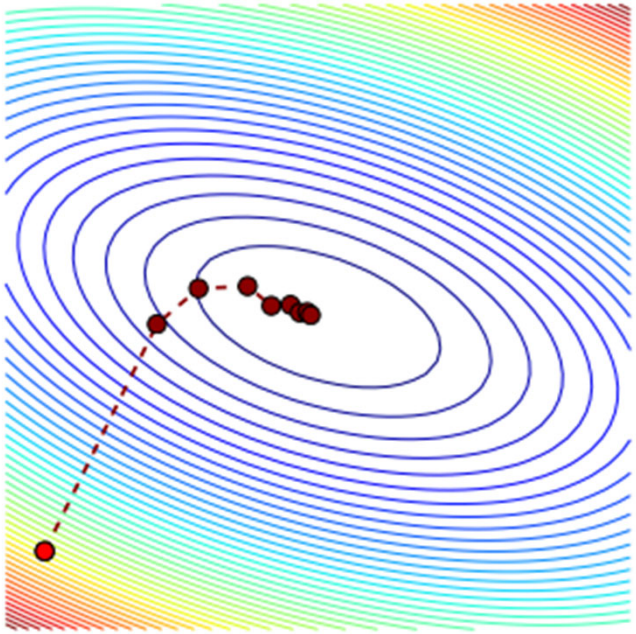
Stochastic Gradient Descent

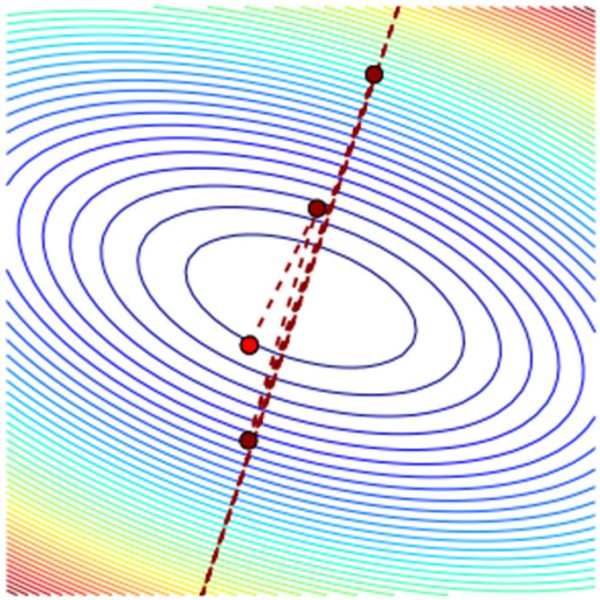
if the function oven or not a linear,
SGD an help with the choosing other data

- initialize $w = 0$
- repeat
  - randomly choose training point $(x_i, y_i) \in \mathcal{D}_n$
  - $w \leftarrow w - \eta \underbrace{\nabla_w \ell(f_w(x_i), y_i)}_{\text{Grad(Loss on i'th example)}}$

Convergence Analysis?

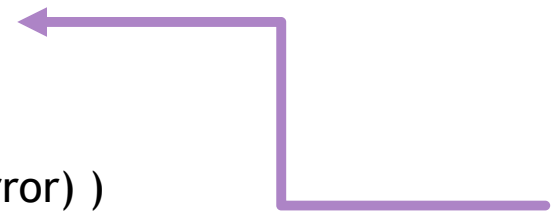






Convergence Analysis?

- Fixed Step Size
  - Learning rate 1/(max of derivative) rate is too slow
  - 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))



LECTURE 7!

Perceptron Algorithm as SGD

PERCEPTRON($\mathbf{w}_0$)

1 $\mathbf{w}_1 \leftarrow \mathbf{w}_0$ ▷ typically $\mathbf{w}_0 = \mathbf{0}$
2 **for** $t \leftarrow 1$ **to** T **do**
3 RECEIVE($\mathbf{x}_t$)
4 $\widehat{y}_t \leftarrow \text{sgn}(\mathbf{w}_t \cdot \mathbf{x}_t)$
5 RECEIVE(y_t)
6 **if** $(\widehat{y}_t \neq y_t)$ **then**
7 $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_t \mathbf{x}_t$ ▷ more generally $\eta y_t \mathbf{x}_t, \eta > 0$.
8 **else** $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t$
9 **return** $\mathbf{w}_{T+1}$

Perceptron Algorithm as SGD

- Take

$$F(\mathbf{w}) = \frac{1}{T} \sum_{t=1}^{T} \max\left(0, -y_t(\mathbf{w} \cdot \mathbf{x}_t)\right)$$

Perceptron Algorithm as SGD

- Take

$$F(\mathbf{w}) = \frac{1}{T}\sum_{t=1}^{T} \max\left(0, -y_t(\mathbf{w}\cdot\mathbf{x}_t)\right)$$

- Set

$$\widetilde{F}(\mathbf{w}, \mathbf{x}) = \max\left(0, -f(\mathbf{x})(\mathbf{w}\cdot\mathbf{x})\right)$$

Perceptron Algorithm as SGD

- Take

$$F(\mathbf{w}) = \frac{1}{T}\sum_{t=1}^{T} \max\left(0, -y_t(\mathbf{w}\cdot\mathbf{x}_t)\right)$$

margin based loss

- Set

$$\widetilde{F}(\mathbf{w},\mathbf{x}) = \max\left(0, -f(\mathbf{x})(\mathbf{w}\cdot\mathbf{x})\right)$$

only the line is wrong can be -f(x)(wx)

- Update Guess

w_t+1 = w_t + y*x

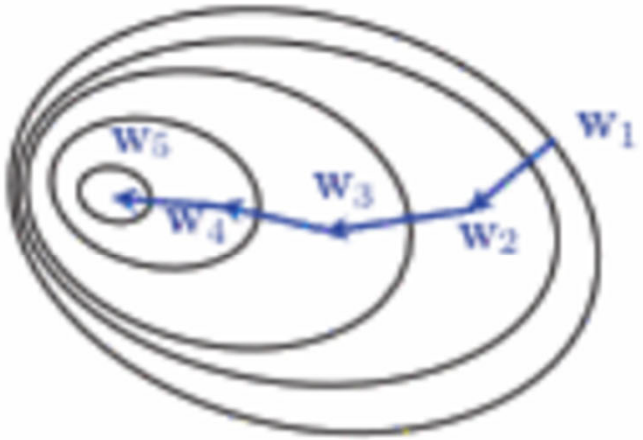
$$\mathbf{w}_{t+1} \leftarrow \begin{cases} \mathbf{w}_t - \eta\nabla_{\mathbf{w}}\widetilde{F}(\mathbf{w}_t,\mathbf{x}_t) & \text{if } \mathbf{w}\mapsto\widetilde{F}(\mathbf{w},\mathbf{x}_t) \text{ differentiable at } \mathbf{w}_t \\ \mathbf{w}_t & \text{otherwise,} \end{cases}$$

F is 0,

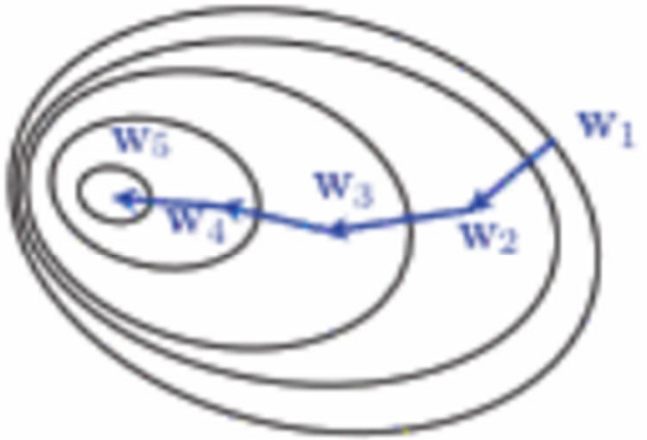
Perceptron Algorithm as SGD

here is the margin

$$\mathbf{w}_{t+1} \leftarrow \begin{cases} \mathbf{w}_t + \eta y_t \mathbf{x}_t & \text{if } y_t(\mathbf{w} \cdot \mathbf{x}_t) < 0; \\ \mathbf{w}_t & \text{if } y_t(\mathbf{w} \cdot \mathbf{x}_t) > 0; \\ \mathbf{w}_t & \text{otherwise,} \end{cases}$$



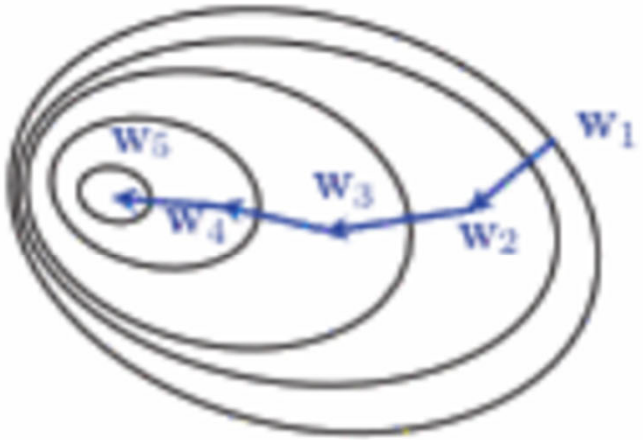
Perceptron Algorithm as SGD



$$\mathbf{w}_{t+1} \leftarrow \begin{cases} \mathbf{w}_t + \eta y_t \mathbf{x}_t & \text{if } y_t(\mathbf{w} \cdot \mathbf{x}_t) < 0; \\ \mathbf{w}_t & \text{if } y_t(\mathbf{w} \cdot \mathbf{x}_t) > 0; \\ \mathbf{w}_t & \text{otherwise,} \end{cases}$$

$$\nabla_{\mathbf{w}} \widetilde{F}(\mathbf{w}, \mathbf{x}_t) = -y\mathbf{x}_t \quad \text{if} \quad y_t(\mathbf{w} \cdot \mathbf{x}_t) < 0$$

Perceptron Algorithm as SGD



$$\mathbf{w}_{t+1} \leftarrow \begin{cases} \mathbf{w}_t + \eta y_t \mathbf{x}_t & \text{if } y_t(\mathbf{w} \cdot \mathbf{x}_t) < 0; \\ \mathbf{w}_t & \text{if } y_t(\mathbf{w} \cdot \mathbf{x}_t) > 0; \\ \mathbf{w}_t & \text{otherwise,} \end{cases}$$

$$\nabla_{\mathbf{w}} \widetilde{F}(\mathbf{w}, \mathbf{x}_t) = -y\mathbf{x}_t \quad \text{if} \quad y_t(\mathbf{w} \cdot \mathbf{x}_t) < 0$$

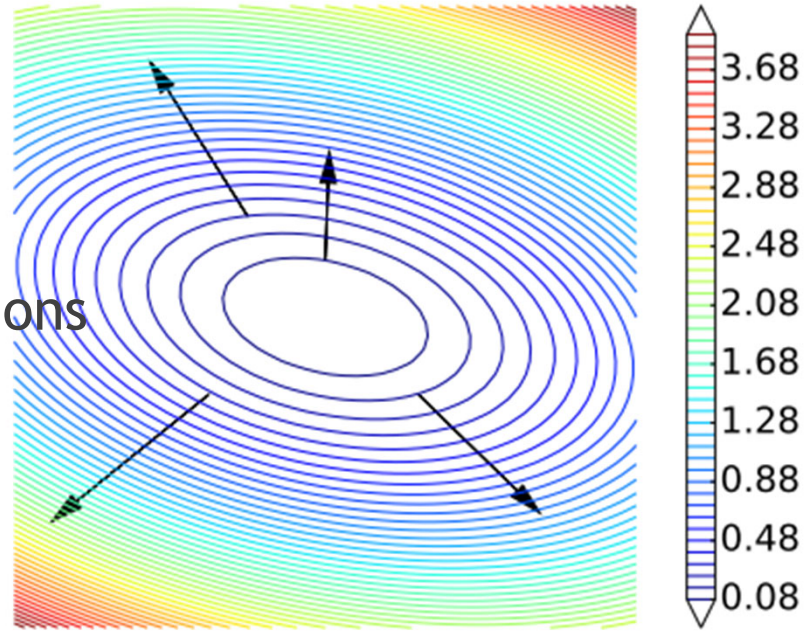
$$\nabla_{\mathbf{w}} \widetilde{F}(\mathbf{w}, \mathbf{x}_t) = 0 \text{ if } y_t(\mathbf{w} \cdot \mathbf{x}_t) > 0.$$

Objectives

- Overfitting
  - How to avoid overfitting by using regularization?
- Regularization
  - Formulate a linear estimation problem with a regularization
  - Compute an L1-regularized or L2-regularized estimate
  - Compute optimal regularization level using cross validation
- Compare and Contrast

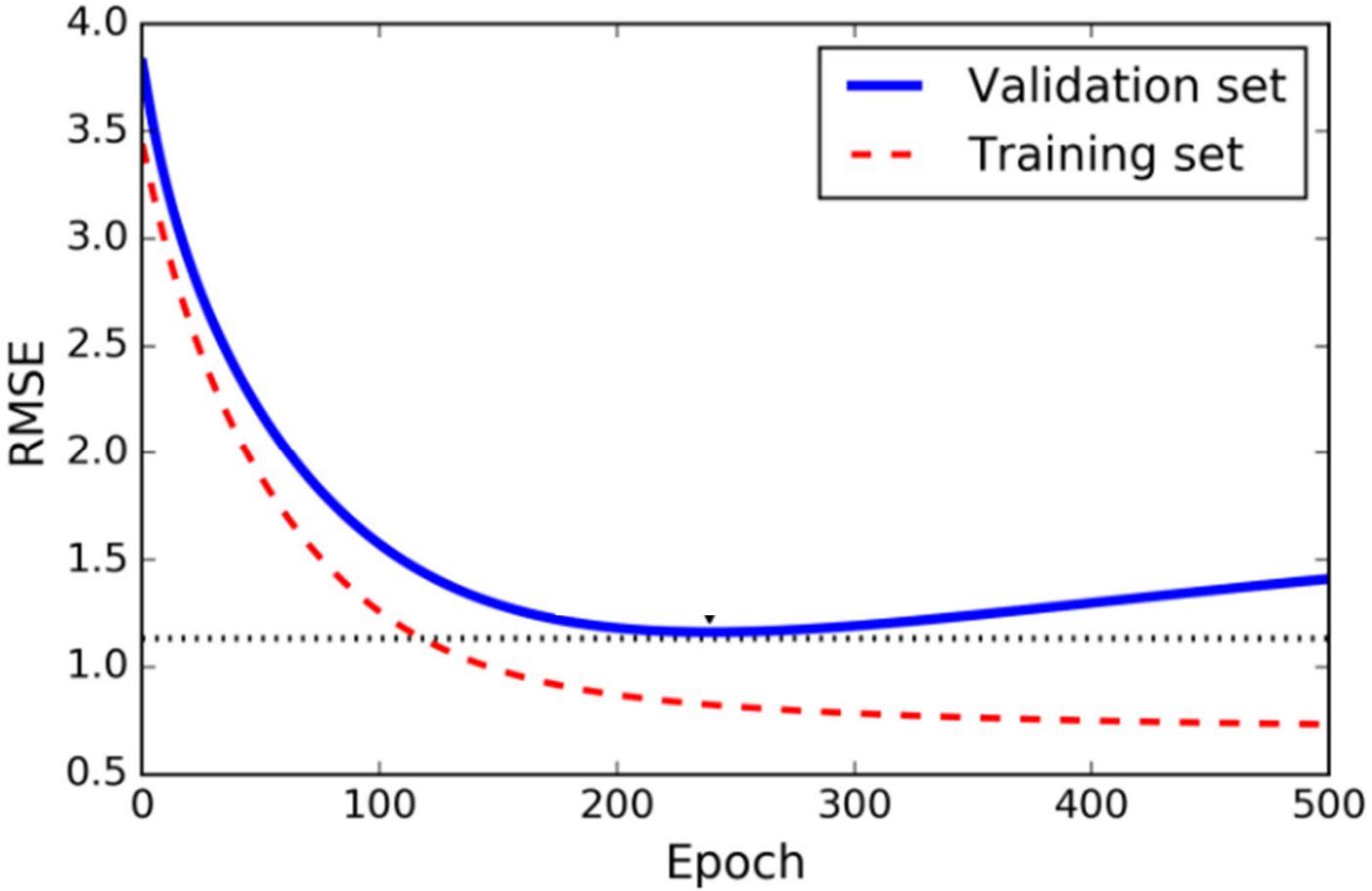
Agenda

- Lesson
  - Ridge Regression and Normal Equations
  - Lasso Regression and Sparsity
  - Similarities and Differences
- Demo
  - Implementing Ridge, Lasso and Elastic Net

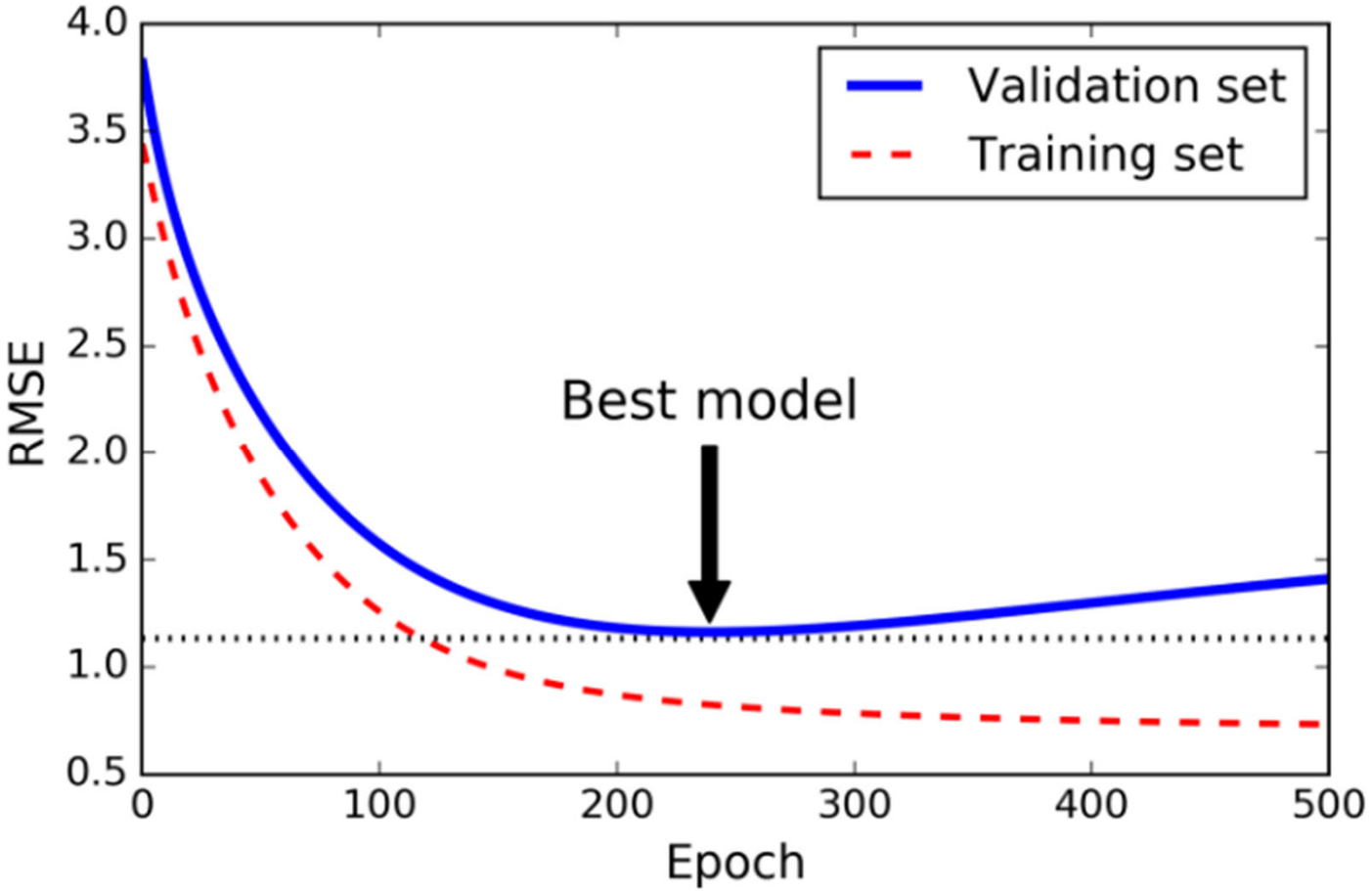


early stoping
RMSE rout mean square error

Why Regularize Loss Functions?



Why Regularize Loss Functions?



Ridge Regression and Normal Equations

$$\frac{1}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w}) = \frac{1}{2}\mathbf{w}^T(\mathbf{X}^T\mathbf{X})\mathbf{w} - \mathbf{w}^T(\mathbf{X}^T\mathbf{y})$$

drive w

$$\mathbf{X}^T\mathbf{X} = \sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T = \sum_{i=1}^{N}\begin{pmatrix} x_{i,1}^2 & \cdots & x_{i,1}x_{i,D} \\ & \ddots & \\ x_{i,D}x_{i,1} & \cdots & x_{i,D}^2 \end{pmatrix}$$

$$\mathbf{X}^T\mathbf{y} = \sum_{i=1}^{N}\mathbf{x}_i y_i.$$

Ridge Regression and Normal Equations

$$\frac{1}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w}) = \frac{1}{2}\mathbf{w}^T(\mathbf{X}^T\mathbf{X})\mathbf{w} - \mathbf{w}^T(\mathbf{X}^T\mathbf{y})$$

- driv w

$$[\mathbf{X}^T\mathbf{X}\mathbf{w} - \mathbf{X}^T\mathbf{y}] = \sum_{i=1}^{N}\mathbf{x}_i(\mathbf{w}^T\mathbf{x}_i - y_i)$$

to solve and minimize the former formule then compute the w

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

Ridge Regression and Normal Equations

$$\frac{1}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w}) = \frac{1}{2}\mathbf{w}^T(\mathbf{X}^T\mathbf{X})\mathbf{w} - \mathbf{w}^T(\mathbf{X}^T\mathbf{y})$$

$$[\mathbf{X}^T\mathbf{X}\mathbf{w} - \mathbf{X}^T\mathbf{y}] = \sum_{i=1}^{N}\mathbf{x}_i(\mathbf{w}^T\mathbf{x}_i - y_i)$$

$$\hat{\mathbf{w}}_{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

ordinary loss solution

How to Understand the Normal Equations?

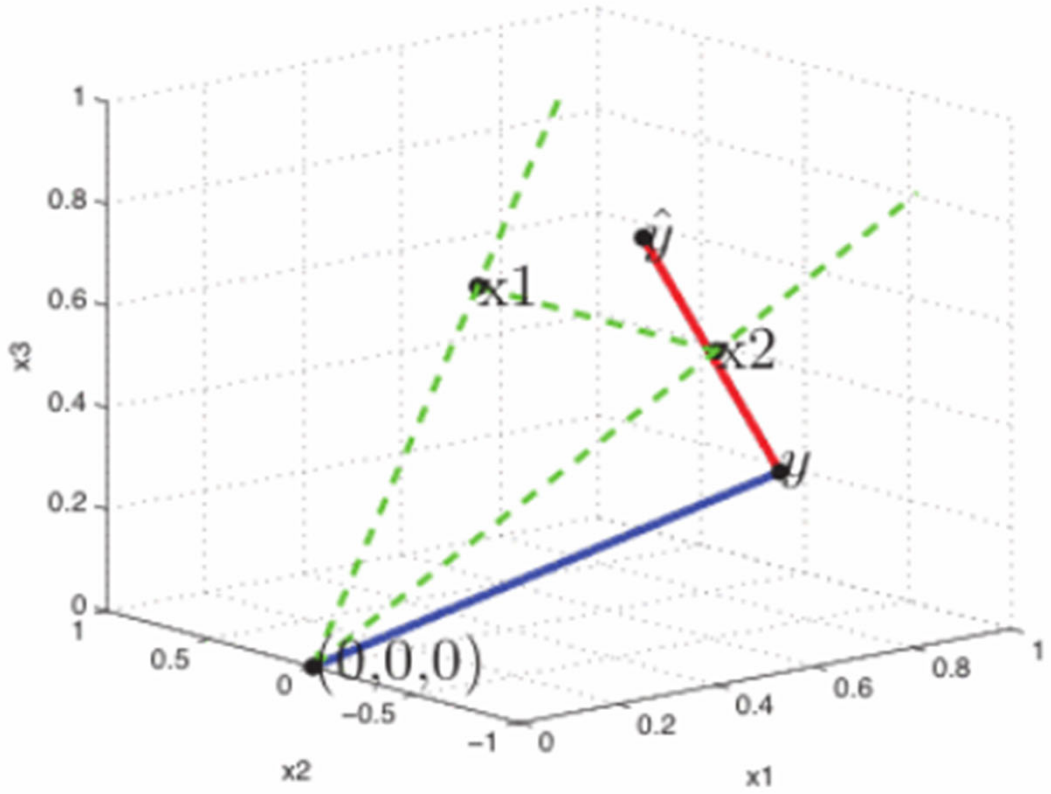
$$\mathbf{X} = \begin{pmatrix} 1 & 2 \\ 1 & -2 \\ 1 & 2 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} 8.8957 \\ 0.6130 \\ 1.7761 \end{pmatrix}$$

$$\hat{\mathbf{y}} = w_1 \tilde{\mathbf{x}}_1 + \cdots + w_D \tilde{\mathbf{x}}_D = \mathbf{X}\mathbf{w}$$

$$\underset{\hat{\mathbf{y}} \in \text{span}(\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_D\})}{\text{argmin}} \|\mathbf{y} - \hat{\mathbf{y}}\|_2.$$

norm

How to Understand the Normal Equations?



$$\tilde{\mathbf{x}}_j^T(\mathbf{y}-\hat{\mathbf{y}}) = 0 \Rightarrow \mathbf{X}^T(\mathbf{y}-\mathbf{X}\mathbf{w}) = \mathbf{0} \Rightarrow \mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Why does Ridge Regression shrink weights?

large weight complexity

as small as possible

bias

$$J(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (w_0 + \mathbf{w}^T\mathbf{x}_i))^2 + \lambda||\mathbf{w}||_2^2$$

w_1^2 + w_2 ^2 + ...

$$\hat{\mathbf{w}}_{ridge} = (\lambda\mathbf{I}_D + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

why to reduce the weight: not to 0

(lambda*I + X.T*X) v = 0

V.T (lambda*I + X.T*X) v = 0

v.T *v*lambda greater than 0

v.T*x.t * x *v <= 0

Why does Ridge Regression shrink weights?

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \longrightarrow \hat{\mathbf{w}}_{ridge} = \mathbf{V}(\mathbf{S}^2 + \lambda\mathbf{I})^{-1}\mathbf{S}\mathbf{U}^T\mathbf{y}$$

Why does Ridge Regression shrink weights?

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \longrightarrow \hat{\mathbf{w}}_{ridge} = \mathbf{V}(\mathbf{S}^2 + \lambda\mathbf{I})^{-1}\mathbf{S}\mathbf{U}^T\mathbf{y}$$

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{X}\hat{\mathbf{w}}_{ridge} = \mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}(\mathbf{S}^2 + \lambda\mathbf{I})^{-1}\mathbf{S}\mathbf{U}^T\mathbf{y} \\ &= \mathbf{U}\tilde{\mathbf{S}}\mathbf{U}^T\mathbf{y} = \sum_{j=1}^{D} \mathbf{u}_j \tilde{S}_{jj} \mathbf{u}_j^T \mathbf{y} \end{aligned}$$

$$\tilde{S}_{jj} \triangleq [\mathbf{S}(\mathbf{S}^2 + \lambda I)^{-1}\mathbf{S}]_{jj} = \frac{\sigma_j^2}{\sigma_j^2 + \lambda}$$

Why does Ridge Regression shrink weights?

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{w}}_{ridge} = \sum_{j=1}^{D} \mathbf{u}_j \frac{\sigma_j^2}{\sigma_j^2 + \lambda} \mathbf{u}_j^T \mathbf{y}$$

VS

the reason that ridge can reduce the weight. lambda increase the weight decrease

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{w}}_{ls} = (\mathbf{U}\mathbf{S}\mathbf{V}^T)(\mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T\mathbf{y}) = \mathbf{U}\mathbf{U}^T\mathbf{y} = \sum_{j=1}^{D} \mathbf{u}_j \mathbf{u}_j^T \mathbf{y}$$

$$\mathrm{dof}(\lambda) = \sum_{j=1}^{D} \frac{\sigma_j^2}{\sigma_j^2 + \lambda}$$

Question

- 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.

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

where

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

Question

- 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.

Question

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

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

Question

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

residual

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

Question

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

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

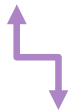
Regularization gives Stability

Ridge Regression (Tikhonov Form)

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

$$\hat{w} = \underset{w \in \mathbf{R}^d}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left\{ w^T x_i - y_i \right\}^2 + \lambda \|w\|_2^2,$$

where $\|w\|_2^2 = w_1^2 + \cdots + w_d^2$ is the square of the ℓ_2-norm.



Ridge Regression (Ivanov Form)

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

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

Regularization gives Stability

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

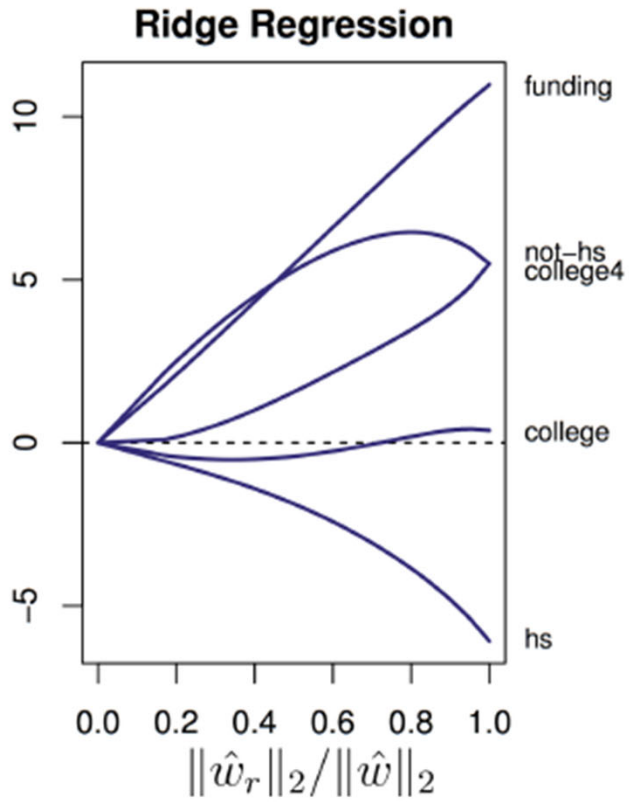
$$\begin{aligned} \left|\hat{f}(x+h) - \hat{f}(x)\right| &= |\hat{w}^T(x+h) - \hat{w}^T x| = \left|\hat{w}^T h\right| \\ &\leqslant \|\hat{w}\|_2 \|h \end{aligned}$$

y)

Question

$$\|\hat{w}\|_1 \geqslant \|\hat{w}\|_2,$$

Regularization Path



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

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

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

Lasso Regression

Lasso Regression (Tikhonov Form)

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

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

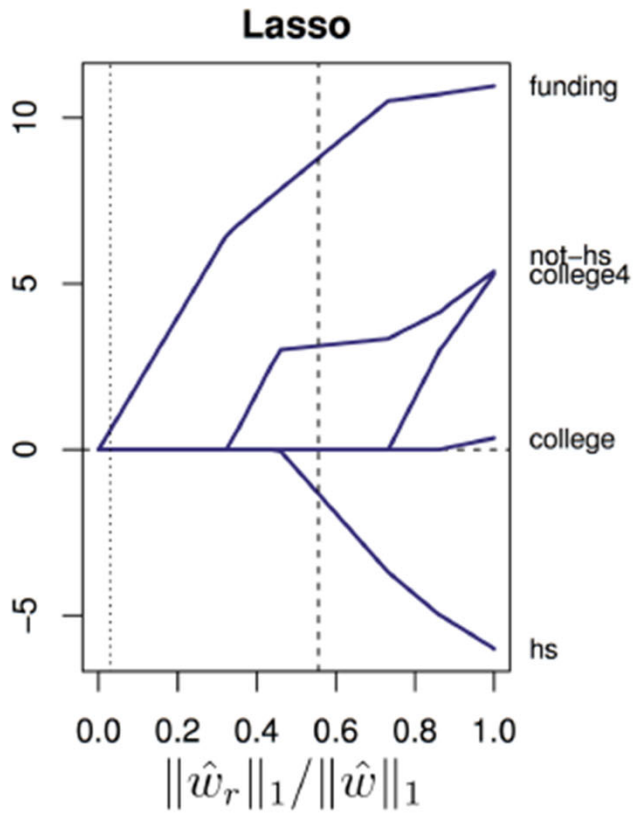
where $\|w\|_1 = |w_1| + \cdots + |w_d|$ is the ℓ_1-norm.

Lasso Regression (Ivanov Form)

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

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

Lasso Regression



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

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

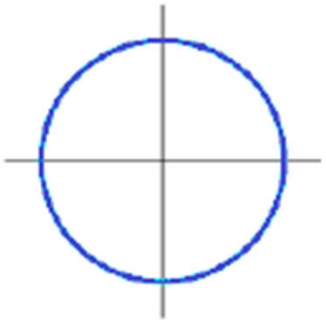
- For $r = 0$, $\|\hat{w}_r\|_1/\|\hat{w}\|_1 = 0$.
- For $r = \infty$, $\|\hat{w}_r\|_1/\|\hat{w}\|_1 = 1$

Lasso Regression and Sparsity

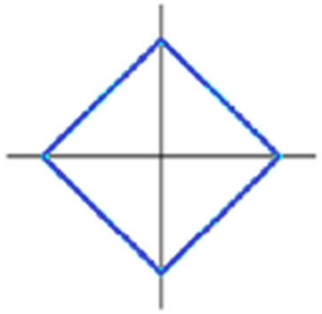
- Coefficient are 0 = > don't need those features. What's the gain?
  - Time/expense to compute/buy features
  - Memory to store features (e.g. real-time deployment)
  - Identifies the important features
  - Better prediction? sometimes
  - As a feature-selection step for training a slower non-linear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

Why Sparse Solutions

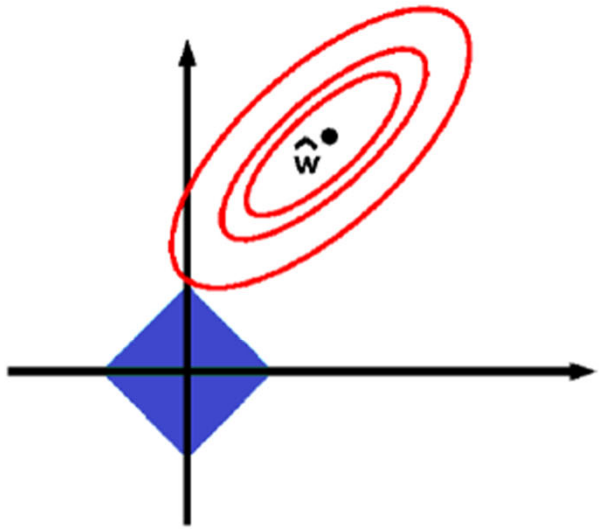
- ℓ_2 contour:
  $w_1^2 + w_2^2 = r$



- ℓ_1 contour:
  $|w_1| + |w_2| = r$



Why Sparse Solutions



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

Why Sparse Solutions

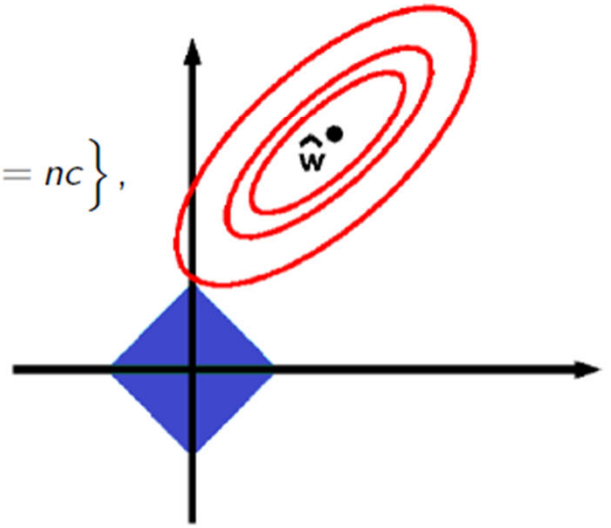
- By "completing the square", we can show for any $w \in \mathbf{R}^d$:

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

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

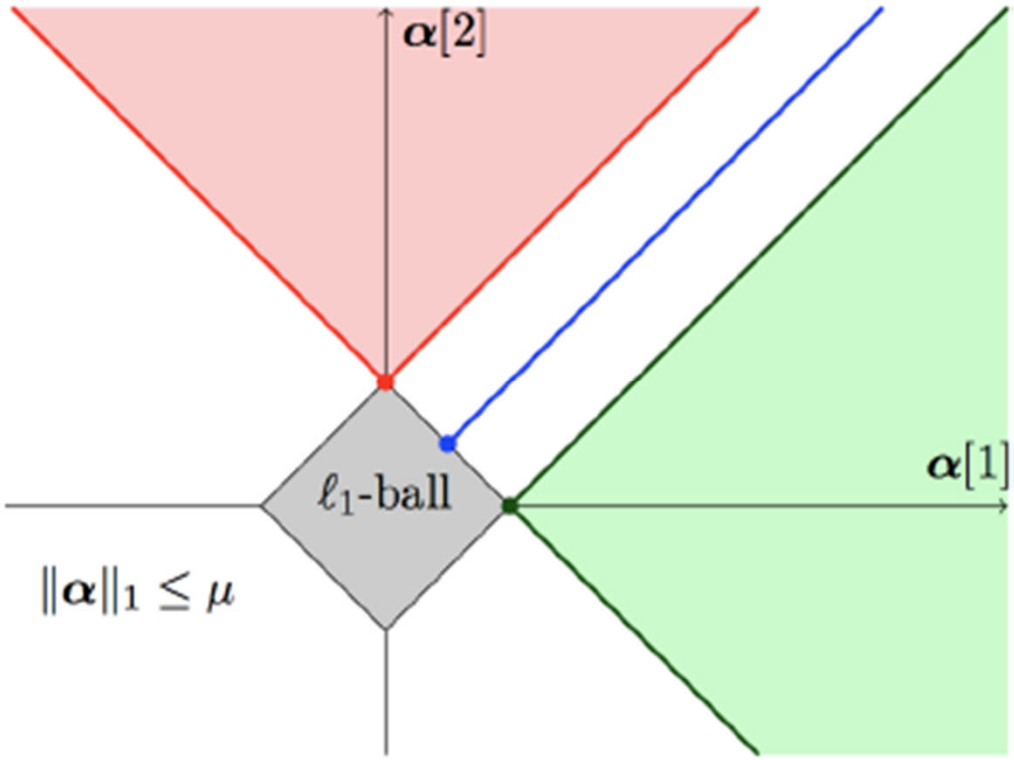
$$\left\{w \mid \hat{R}_n(w) = c + \hat{R}_n(\hat{w})\right\} = \left\{w \mid (w-\hat{w})^T X^T X (w-\hat{w}) = nc\right\},$$

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

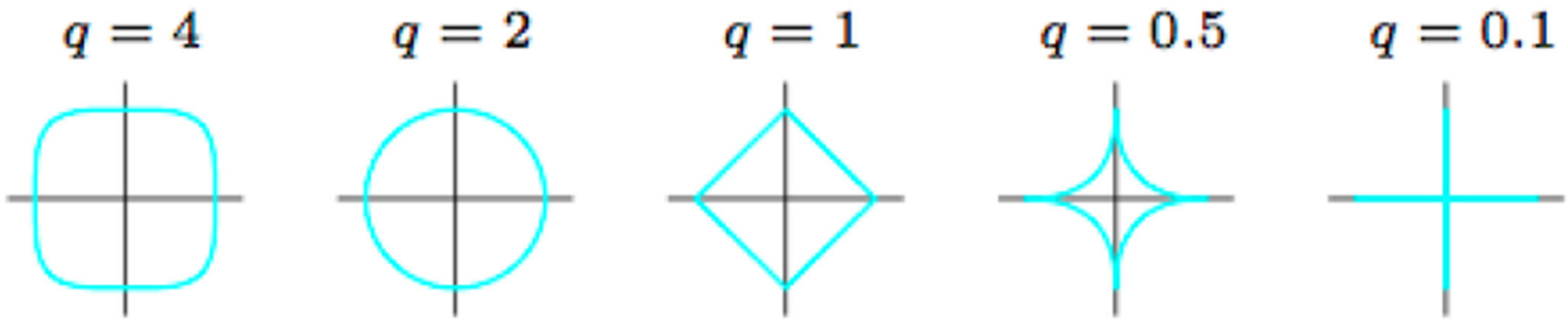


- Blue region: Area satisfying complexity constraint: $|w_1| + |w_2| \leqslant r$
- Red lines: contours of $\hat{R}_n(w) = \sum_{i=1}^n \left(w^T x_i - y_i\right)^2$.

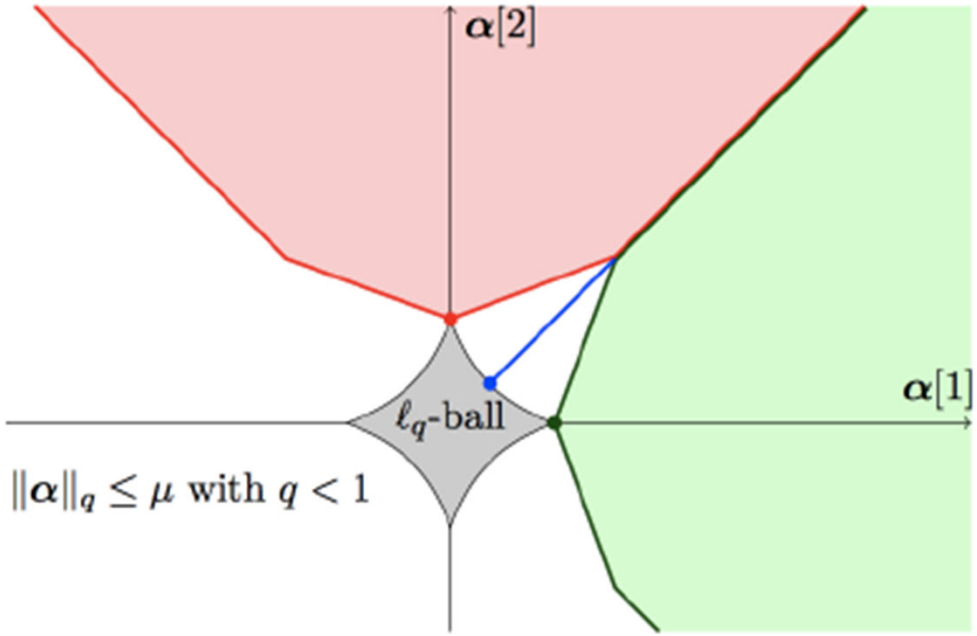
Why Sparse Solutions



Why Sparse Solutions

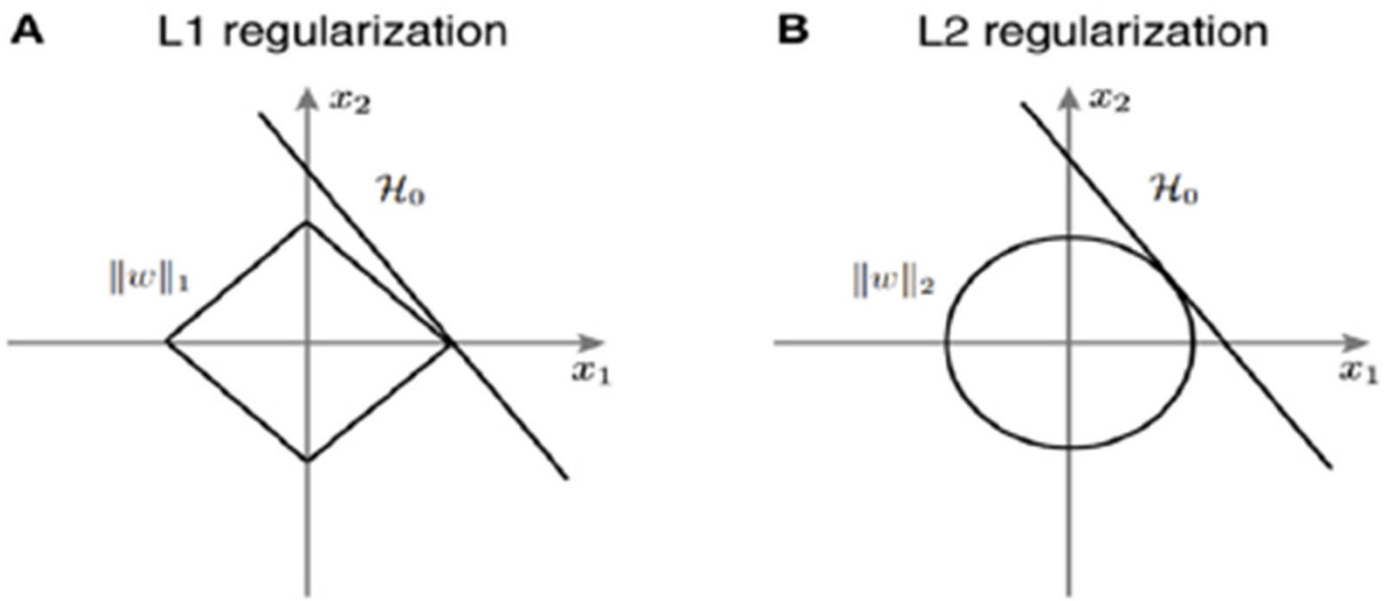


Why Sparse Solutions



(b) ℓ_q-ball with $q < 1$.

Question


A
L1 regularization
x_2
$\mathcal{H}_0$
$\|w\|_1$
x_1
B
L2 regularization
x_2
$\mathcal{H}_0$
$\|w\|_2$
x_1

Take-Aways

- Ridge
  - Explain the normal equations with a picture
  - How can you “invert” a matrix that has a different number of rows and columns
- 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?