Linear Regression

Aarti Singh

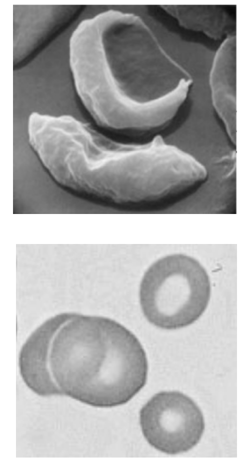
Machine Learning 10-315
Oct 23, 2019

MACHINE LEARNING DEPARTMENT
Carnegie Mellon.
School of Computer Science

Supervised Learning Tasks

Classification

Sports
Science
News



Anemic cell
Healthy cell

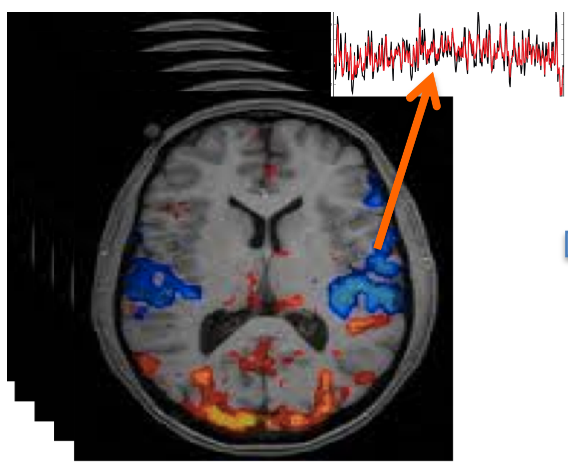
X = Document

Y = Topic

X = Cell Image

Y = Diagnosis

Regression

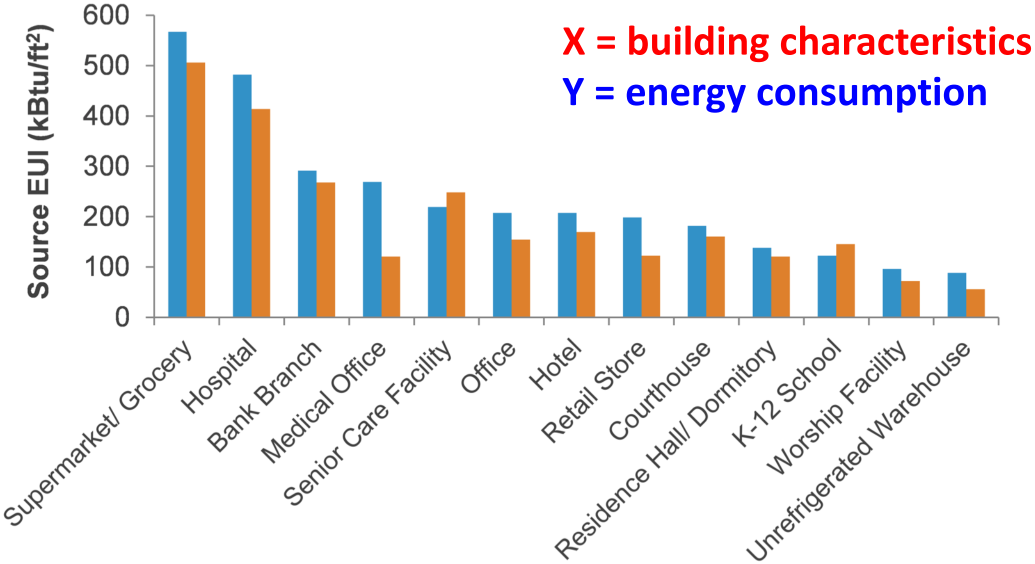


Y = Age of a subject

X = Brain Scan

Regression Tasks

Estimating Energy Usage



energystar.gov

Estimating Contamination

X = new location
Y = sensor reading

Performance Measures

Performance Measure: Quantifies knowledge gained

$\text{loss}(Y, f(X))$ - Measure of closeness between true label *Y* and prediction *f*(*X*)

Don't just want label of one test data (cell image), but any cell image $X \in \mathcal{X}$

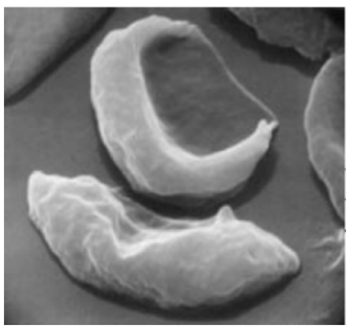
$$(X, Y) \sim P_{XY}$$

Given a cell image drawn randomly from the collection of all cell images, how well does the predictor perform on average?

$$\boxed{\text{Risk } R(f) \equiv \mathbb{E}_{XY}\left[\text{loss}(Y, f(X))\right]}$$

Performance Measures

Performance Measure: Risk $R(f) \equiv \mathbb{E}_{XY}\,[\text{loss}(Y, f(X))]$



“Anemic cell”

Share Price
“$ 24.50”

	loss$(Y, f(X))$	Risk $R(f)$
“Anemic cell”	$1_{\{f(X) \neq Y\}}$ **0/1 loss**	$P(f(X) \neq Y)$ **Probability of Error**
Share Price “$ 24.50”	$(f(X) - Y)^2$ **square loss**	$\mathbb{E}[(f(X) - Y)^2]$ **Mean Square Error**

Bayes Optimal Rule

Ideal goal: Construct **prediction rule** $f^* : \mathcal{X} \to \mathcal{Y}$

$$f^* = \arg\min_f \mathbb{E}_{XY}[\text{loss}(Y, f(X))]$$

Bayes optimal rule

What's the rule for Mean Square Error?
HW3

Best possible performance:

Bayes Risk $R(f^*) \leq R(f)$ for all f

BUT… Optimal rule is not computable - depends on unknown P_{XY} !

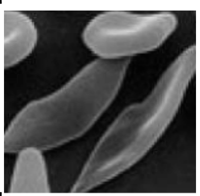
Experience - Training Data

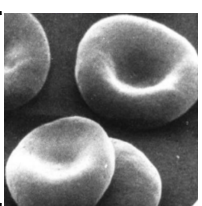
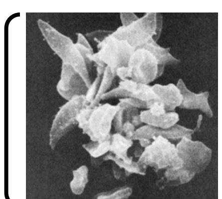
Can't minimize risk since P_{XY} unknown!

Training data (experience) provides a glimpse of P_{XY}

(observed) $\{(X_i, Y_i)\}_{i=1}^{n} \overset{i.i.d.}{\sim} P_{XY}$ **(unknown)**

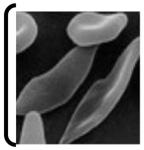
independent, identically distributed

[ , Anemic cell] [ , Healthy cell]

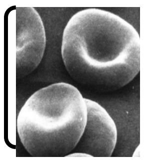
[ , Healthy cell] [ , Anemic cell]

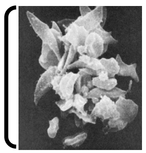
Provided by expert, measuring device, some experiment, …

Machine Learning Algorithm


, Anemic cell


, Healthy cell

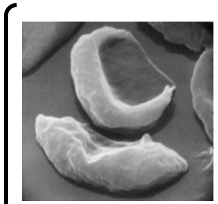

, Healthy cell


, Anemic cell

Learning algorithm $\Rightarrow \widehat{f}_n$

Data $\{(X_i, Y_i)\}_{i=1}^n$

$\widehat{f}_n$ is a mapping from $\mathcal{X} \to \mathcal{Y}$

$\widehat{f}_n$  = "Anemic cell"

Test data X

Empirical Risk Minimization

Optimal predictor: $$f^* = \arg\min_f \mathbb{E}[(f(X) - Y)^2]$$

Empirical Minimizer: $$\widehat{f}_n = \arg\min_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} (f(X_i) - Y_i)^2$$

Empirical mean

Law of Large Numbers:

$$\boxed{\frac{1}{n}\sum_{i=1}^{n} [\text{loss}(Y_i, f(X_i))] \xrightarrow{n \to \infty} \mathbb{E}_{XY}[\text{loss}(Y, f(X))]}$$

Restrict class of predictors

Optimal predictor: $$f^* = \arg\min_f \mathbb{E}[(f(X) - Y)^2]$$

Empirical Minimizer: $$\widehat{f}_n = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2$$

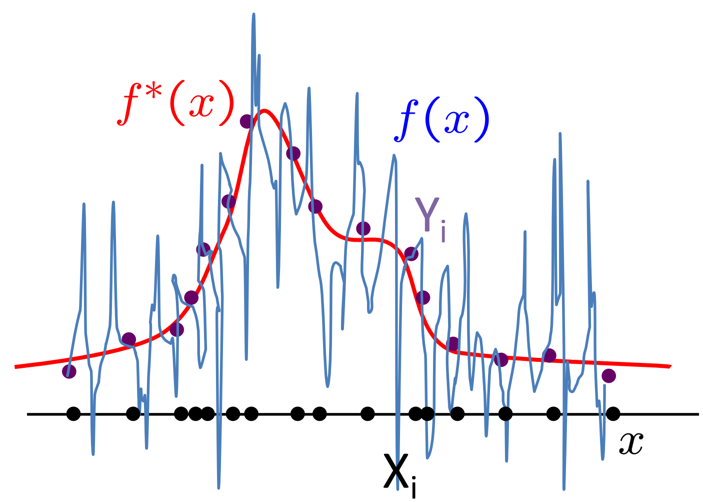
Class of predictors

Why?

Overfitting!

Empiricial loss minimized by any function of the form

$$f(x) = \begin{cases} Y_i, & x = X_i \text{ for } i = 1, \ldots, n \\ \text{any value}, & \text{otherwise} \end{cases}$$



Restrict class of predictors

Optimal predictor:

$$f^* = \arg\min_f \mathbb{E}[(f(X) - Y)^2]$$

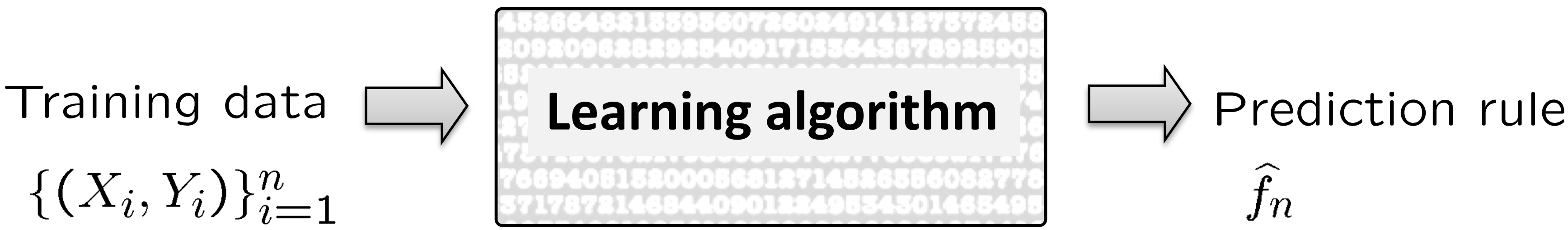
Empirical Minimizer:

$$\widehat{f}_n = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2$$

Class of predictors

$\mathcal{F}$ - Class of Linear functions
- Class of Polynomial functions
- Class of nonlinear functions

Regression algorithms



Linear Regression
Regularized Linear Regression – Ridge regression, Lasso
Polynomial Regression
Kernelized Ridge Regression
Gaussian Process Regression
Kernel regression, Regression Trees, Splines, Wavelet estimators, …

Linear Regression

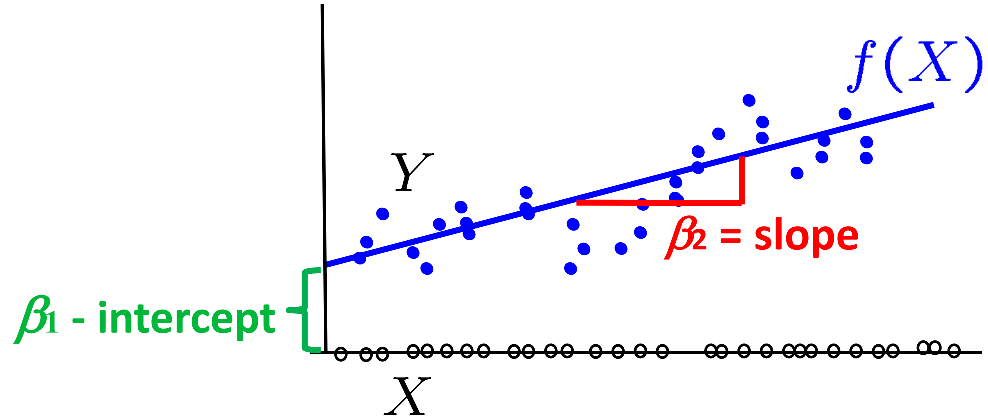
$$\widehat{f}_n^L = \arg\min_{f \in \mathcal{F}_L} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2$$

Least Squares Estimator

$\mathcal{F}_L$ - Class of Linear functions

Uni-variate case:

$$f(X) = \beta_1 + \beta_2 X$$



Multi-variate case:

$$f(X) = f(X^{(1)}, \ldots, X^{(p)}) = \beta_1 \cancel{X^{(1)}}^{\,1} + \beta_2 X^{(2)} + \cdots + \beta_p X^{(p)}$$

$$= X\beta \qquad \text{where} \quad X = [X^{(1)} \ldots X^{(p)}], \quad \beta = [\beta_1 \ldots \beta_p]^T$$

Least Squares Estimator

$$\widehat{f}_n^L = \arg\min_{f \in \mathcal{F}_L} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2 \qquad f(X_i) = X_i \beta$$

$$\Downarrow$$

$$\widehat{\beta} = \arg\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} (X_i \beta - Y_i)^2 \qquad \widehat{f}_n^L(X) = X\widehat{\beta}$$

$$= \arg\min_{\beta} \frac{1}{n} (\mathbf{A}\beta - \mathbf{Y})^T (\mathbf{A}\beta - \mathbf{Y})$$

$$\mathbf{A} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} X_1^{(1)} & \dots & X_1^{(p)} \\ \vdots & \ddots & \vdots \\ X_n^{(1)} & \dots & X_n^{(p)} \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_n \end{bmatrix}$$

Least Squares Estimator

$$\widehat{\beta} = \arg\min_{\beta} \frac{1}{n}(\mathbf{A}\beta - \mathbf{Y})^T(\mathbf{A}\beta - \mathbf{Y}) = \arg\min_{\beta} J(\beta)$$

$$J(\beta) = (\mathbf{A}\beta - \mathbf{Y})^T(\mathbf{A}\beta - \mathbf{Y})$$

$$\left.\frac{\partial J(\beta)}{\partial \beta}\right|_{\widehat{\beta}} = 0$$

Least Square solution satisfies Normal Equations

$$(\mathbf{A}^T\mathbf{A})\widehat{\beta} = \mathbf{A}^T\mathbf{Y}$$

p x p   p x1   p x1

If $(\mathbf{A}^T\mathbf{A})$ is invertible,

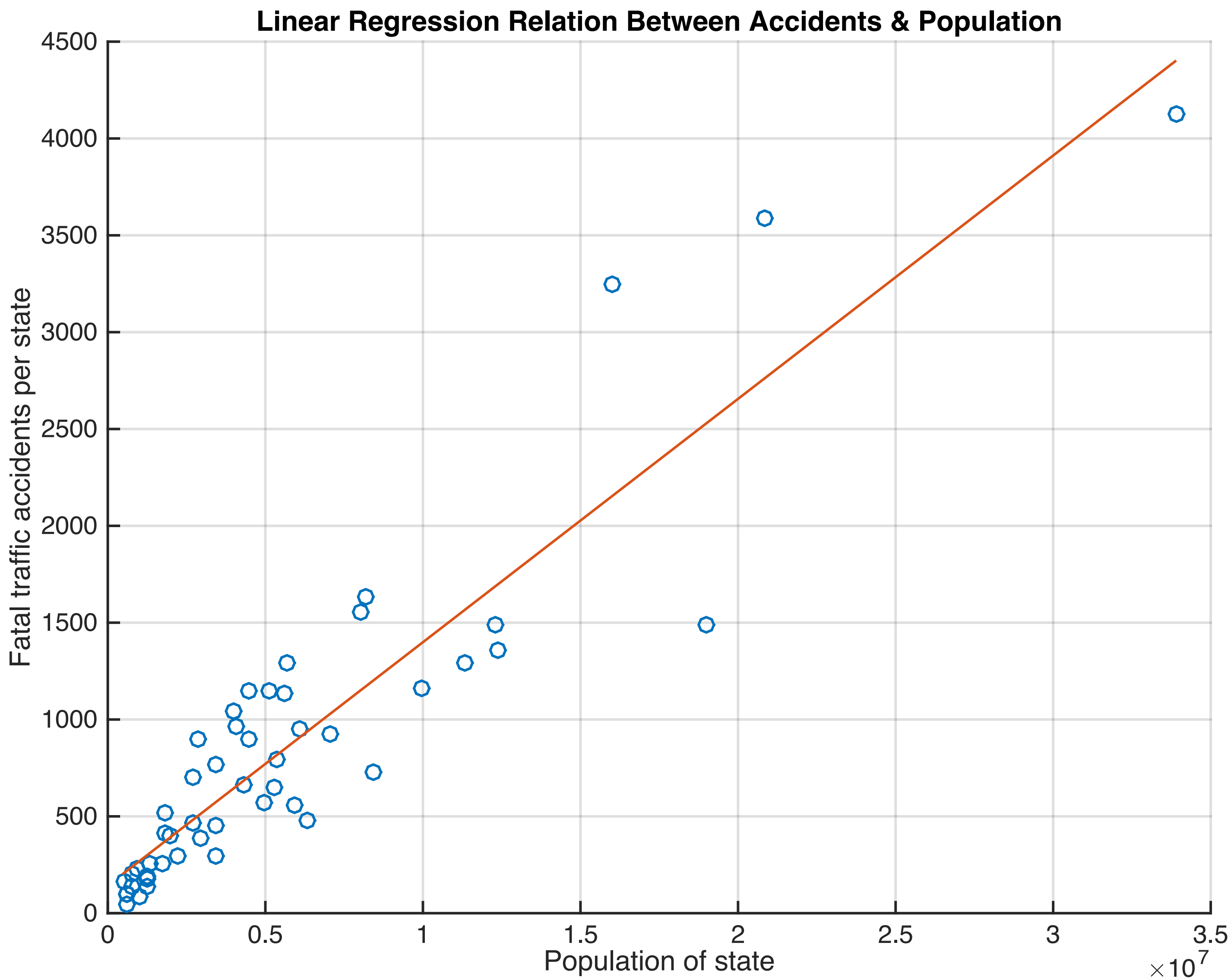
$$\boxed{\widehat{\beta} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Y} \qquad \widehat{f}_n^L(X) = X\widehat{\beta}}$$

Matlab example – linear regression

```
load accidents
x = hwydata(:,14);                  %Population of states
y = hwydata(:,4);                   %Accidents per state
scatter(x,y)
hold on
X = [ones(length(x),1) x];

b = X\y;
yhat = X*b;
plot(x,yhat)
xlabel('Population of state')
ylabel('Fatal traffic accidents per state')
title('Linear Regression Relation Between Accidents &
Population')
```

Matlab example – linear regression



Least Square solution satisfies Normal Equations

$$(\mathbf{A}^T\mathbf{A})\widehat{\beta} = \mathbf{A}^T\mathbf{Y}$$

p x p p x1 p x1

If $(\mathbf{A}^T\mathbf{A})$ is invertible,

$$\boxed{\widehat{\beta} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Y} \qquad \widehat{f}_n^L(X) = X\widehat{\beta}}$$

Later: When is $(\mathbf{A}^T\mathbf{A})$ invertible ?
Recall: Full rank matrices are invertible. What is rank of $(\mathbf{A}^T\mathbf{A})$?

Now: What if $(\mathbf{A}^T\mathbf{A})$ is invertible but expensive (p very large)?

Gradient Descent

Even when $(\mathbf{A}^T\mathbf{A})$ is invertible, might be computationally expensive if **A** is huge.

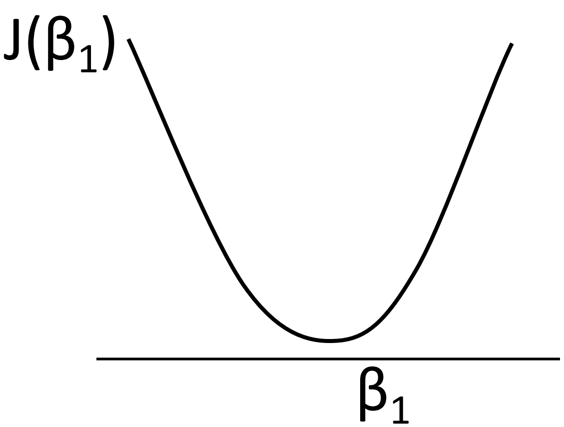
$$\widehat{\beta} = \arg\min_{\beta} \frac{1}{n}(\mathbf{A}\beta - \mathbf{Y})^T(\mathbf{A}\beta - \mathbf{Y}) = \arg\min_{\beta} J(\beta)$$

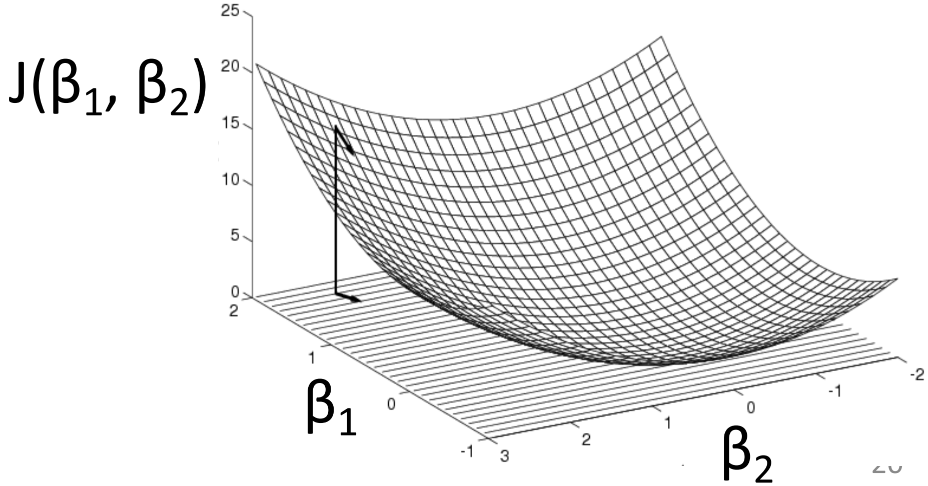
Treat as optimization problem

Why? HW3

<u>Observation:</u> J(β) is convex in β.

How to find the minimizer?



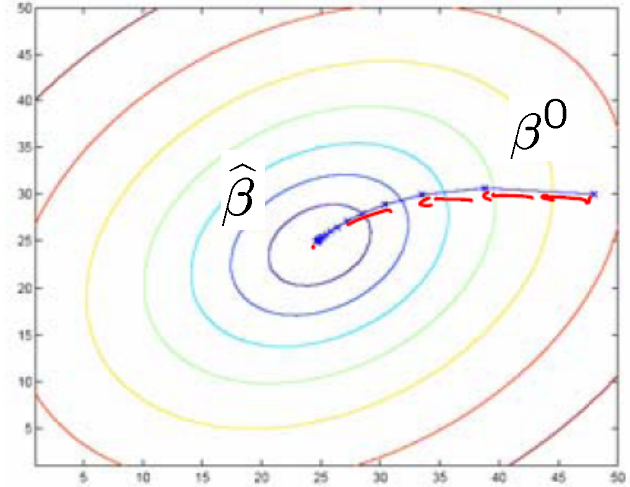


Gradient Descent

Even when $(\mathbf{A}^T\mathbf{A})$ is invertible, might be computationally expensive if **A** is huge.

$$\widehat{\beta} = \arg\min_{\beta} \frac{1}{n}(\mathbf{A}\beta - \mathbf{Y})^T(\mathbf{A}\beta - \mathbf{Y}) = \arg\min_{\beta} J(\beta)$$

Since J(β) is convex, move along negative of gradient



Initialize: β^0

step size

Update:
$$\beta^{t+1} = \beta^t - \frac{\alpha}{2}\frac{\partial J(\beta)}{\partial \beta}\Big|_t = \beta^t - \alpha\ \mathbf{A}^T\underbrace{(\mathbf{A}\beta^t - Y)}_{0 \text{ if } \widehat{\beta} = \beta^t}$$

Stop: when some criterion met e.g. fixed # iterations, or $\frac{\partial J(\beta)}{\partial \beta}\Big|_{\beta^t} < \varepsilon$.