Linear Regression

Aarti Singh

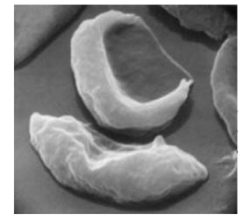
Machine Learning 10-701/15-781
Sept 27, 2010

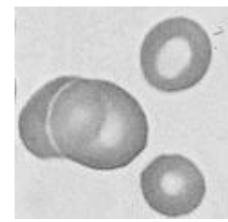
MACHINE LEARNING DEPARTMENT
Carnegie Mellon.
School of Computer Science

Discrete to Continuous Labels

Classification

Sports
Science
News



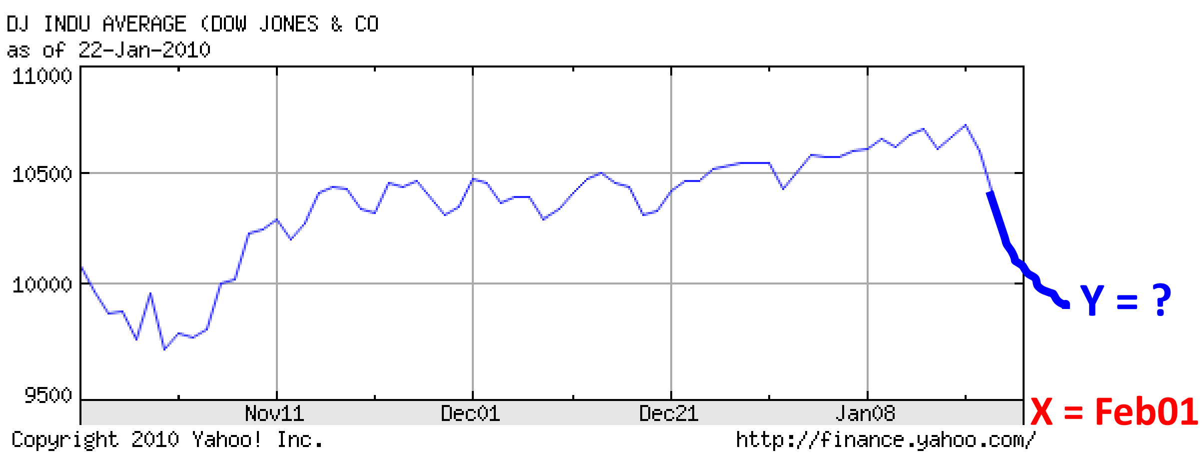


Anemic cell
Healthy cell

X = Document **Y = Topic** **X = Cell Image** **Y = Diagnosis**

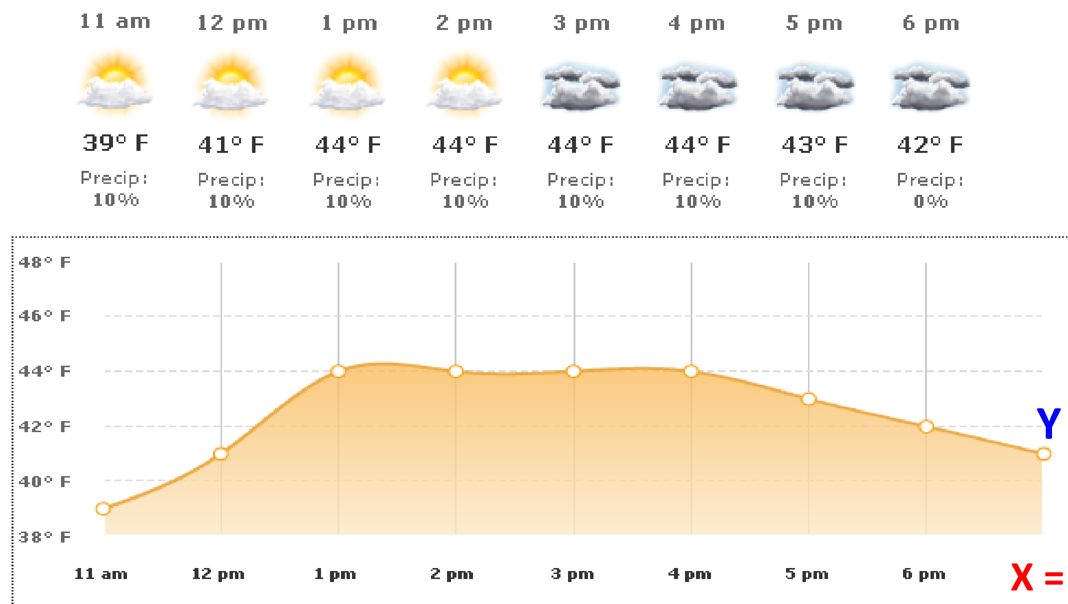
Regression

Stock Market Prediction



Regression Tasks

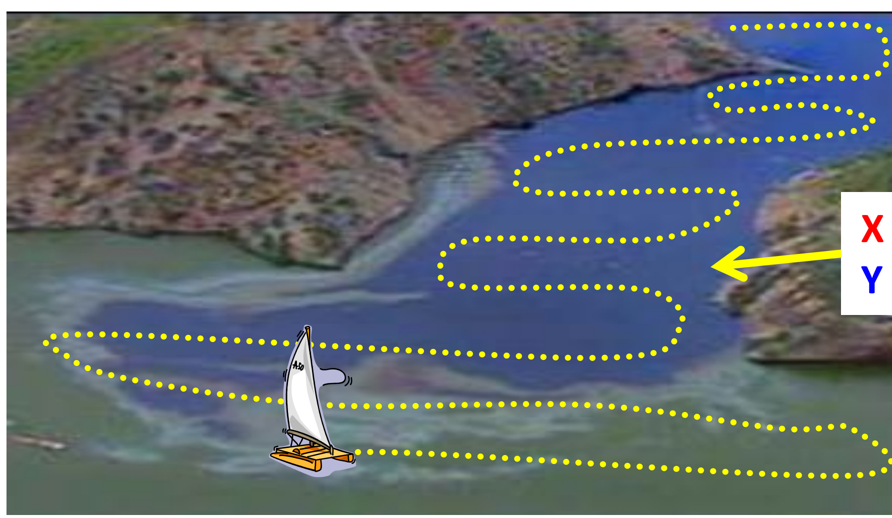
Weather Prediction



Y = Temp

X = 7 pm

Estimating Contamination



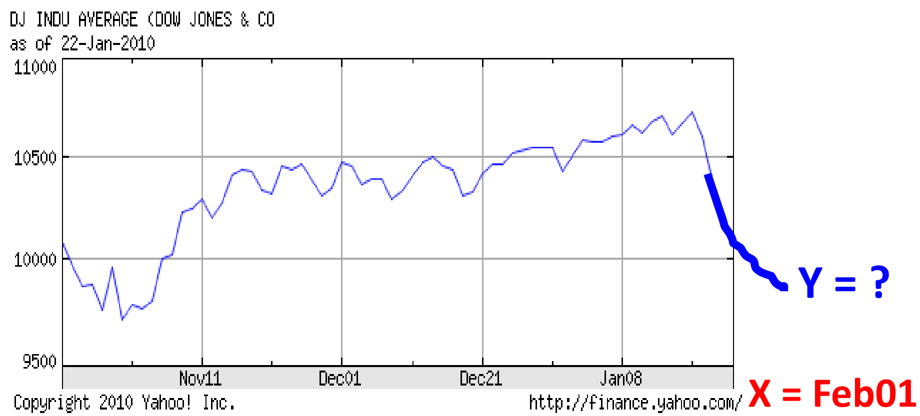
X = new location

Y = sensor reading

Supervised Learning

Goal: Construct a **predictor** $f : X \to Y$ to minimize a risk (performance measure) $R(f)$

Sports
Science
News



Classification:

$$R(f) = P(f(X) \neq Y)$$

Probability of Error

Regression:

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

Mean Squared Error

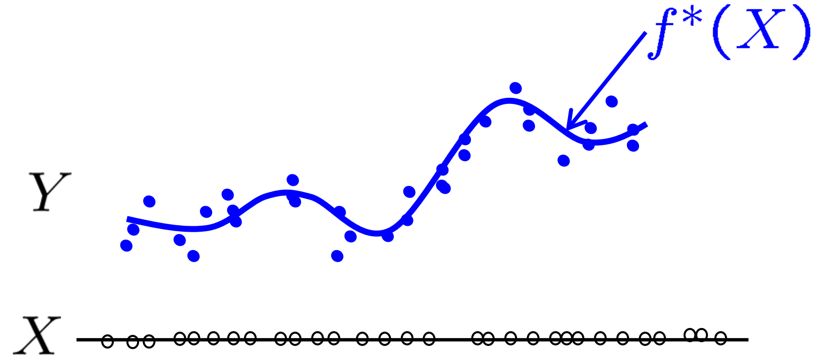
Regression

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

$$= \mathbb{E}[Y|X]$$ (Conditional Mean)

Intuition: Signal plus (zero-mean) Noise model

$$Y = f^*(X) + \epsilon$$



Regression

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

Proof Strategy: $R(f) \geq R(f^*)$ for any prediction rule f

$$R(f) = \mathbb{E}_{XY}[(f(X) - Y)^2] = \mathbb{E}_X[\mathbb{E}_{Y|X}[(f(X) - Y)^2|X]]$$

Dropping subscripts for notational convenience

$$= E\left[E\left[(\underbrace{f(X) - E[Y|X]} + \underbrace{E[Y|X] - Y})^2|X\right]\right]$$

$$= E\big[\; E[(f(X) - E[Y|X])^2|X] + 2E\left[(f(X) - E[Y|X])(E[Y|X] - Y)|X\right] + E[(E[Y|X] - Y)^2|X]\big]$$

$$= E\big[\; E[(f(X) - E[Y|X])^2|X] + 2(f(X) - E[Y|X]) \times 0 + E[(E[Y|X] - Y)^2|X]\big]$$

$$= \underbrace{E\left[(f(X) - E[Y|X])^2\right]}_{\geq 0} + R(f^*).$$

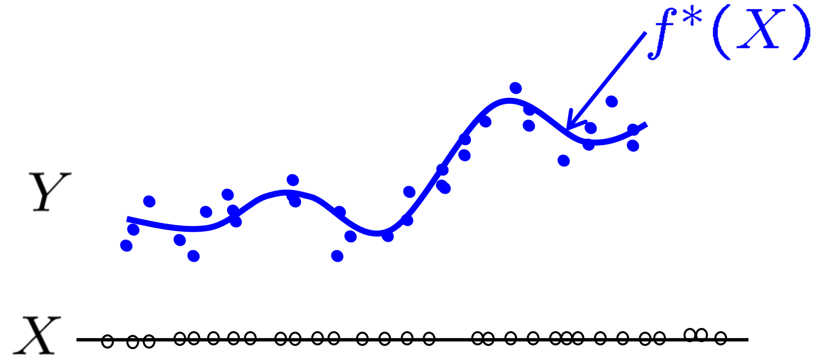
Regression

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

$$= \mathbb{E}[Y|X]$$ (Conditional Mean)

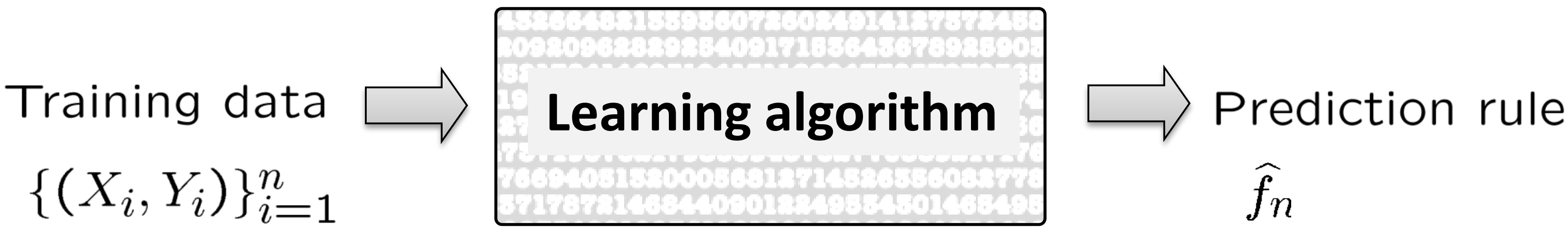
Intuition: Signal plus (zero-mean) Noise model

$$Y = f^*(X) + \epsilon$$



Depends on **unknown** distribution P_{XY}

Regression algorithms



Linear Regression

Lasso, Ridge regression (Regularized Linear Regression)

Nonlinear Regression

Kernel Regression

Regression Trees, Splines, Wavelet estimators, …

Empirical Risk Minimization (ERM)

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

Empirical Risk 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 **Empirical mean**

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

More later…

ERM – you saw it before!

- Learning Distributions

  Max likelihood = Min -ve log likelihood empirical risk

$$\max_\theta P(D|\theta) = \min_\theta \frac{1}{n}\sum_{i=1}^{n} \underbrace{-\log P(X_i|\theta)}_{\text{loss}(X_i,\theta)}$$

Negative log Likelihood loss

What is the class $\mathcal{F}$?

Class of parametric distributions

Bernoulli (θ)

Gaussian (μ, σ^2)

Linear Regression

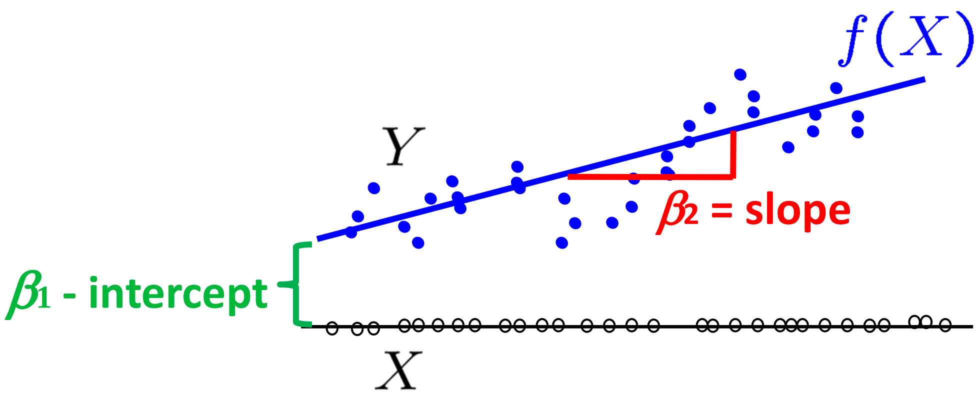
$$\hat{f}_n^L = \arg\min_{f \in \mathcal{F}_L} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2$$

Least Squares Estimator

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

Uni-variate case:

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



Multi-variate case:

1

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

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

Least Squares Estimator

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

⇩

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

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

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

Least Squares Estimator

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

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

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

Normal Equations

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

p xp   p x1   p x1

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

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

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

What if $(\mathbf{A}^T\mathbf{A})$ is not invertible ?
Regularization (later)

Geometric Interpretation

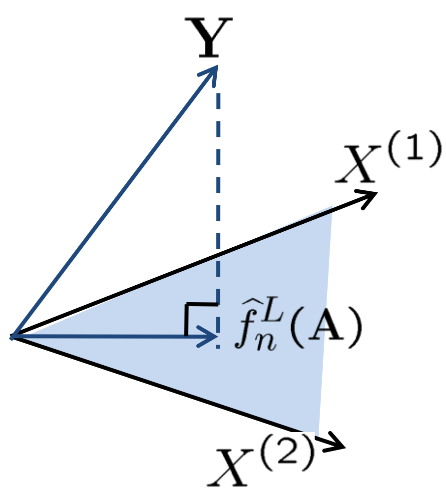
$$\widehat{f}_n^L(X) = X\widehat{\beta} = X(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Y}$$

Difference in prediction on training set:

$$\widehat{f}_n^L(\mathbf{A}) - \mathbf{Y} =$$

$$\mathbf{A}^T(\widehat{f}_n^L(\mathbf{A}) - \mathbf{Y}) = 0$$

$\widehat{f}_n^L(\mathbf{A})$ is the orthogonal projection of $\mathbf{Y}$ onto the linear subspace spanned by the columns of $\mathbf{A}$



Revisiting Gradient Descent

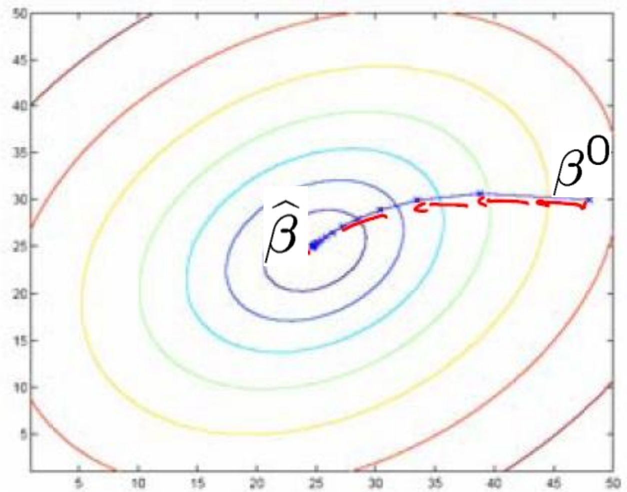
Even when $(\mathbf{A}^T\mathbf{A})$ is invertible, might be computationally expensive if **A** is huge.

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

Gradient Descent since J(β) is convex

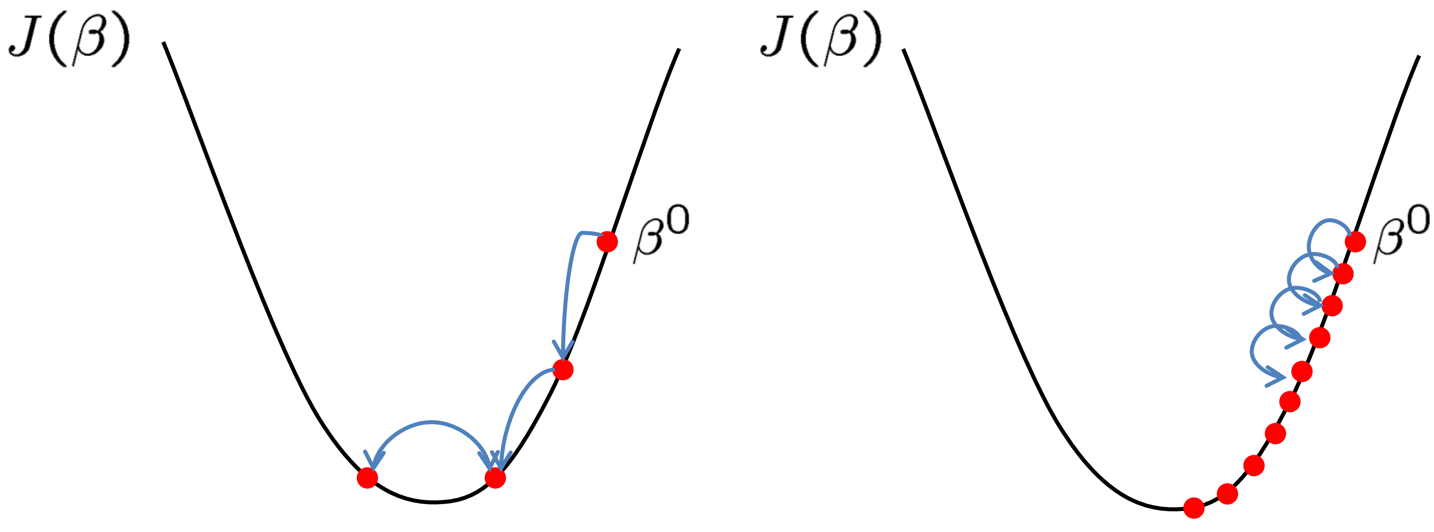
Initialize: β^0

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



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

Effect of step-size α



Large α => Fast convergence but larger residual error
Also possible oscillations

Small α => Slow convergence but small residual error

Least Squares and MLE

Intuition: Signal plus (zero-mean) Noise model

$$Y = f^*(X) + \epsilon = X\beta^* + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$$

$$Y \sim \mathcal{N}(X\beta^*, \sigma^2 \mathbf{I})$$

$$\widehat{\beta}_{\text{MLE}} = \arg\max_{\beta} \underbrace{\log p(\{(X_i, Y_i)\}_{i=1}^{n} | \beta, \sigma^2)}_{\text{log likelihood}}$$

$$= \arg\min_{\beta} \sum_{i=1}^{n} (X_i\beta - Y_i)^2 = \widehat{\beta}$$

Least Square Estimate is same as Maximum Likelihood Estimate under a Gaussian model !

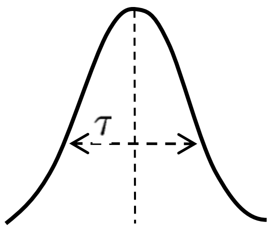
Regularized Least Squares and MAP

What if $(\mathbf{A}^T\mathbf{A})$ is not invertible ?

$$\widehat{\beta}_{\text{MAP}} = \arg\max_{\beta} \underbrace{\log p(\{(X_i, Y_i)\}_{i=1}^n | \beta, \sigma^2)}_{\text{log likelihood}} + \underbrace{\log p(\beta)}_{\text{log prior}}$$

I) Gaussian Prior

$$\beta \sim \mathcal{N}(0, \tau^2 \mathbf{I}) \qquad p(\beta) \propto e^{-\beta^T\beta/2\tau^2}$$



$$\widehat{\beta}_{\text{MAP}} = \arg\min_{\beta} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \lambda \|\beta\|_2^2$$

Ridge Regression

$\lambda \rightarrow$ constant(σ^2, τ^2)

Closed form: HW

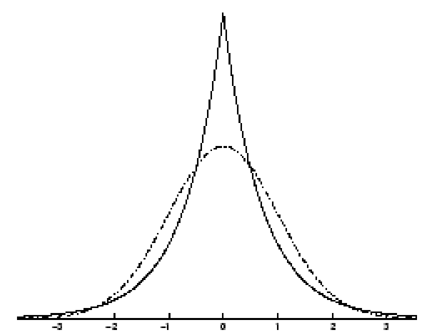
Prior belief that β is Gaussian with zero-mean biases solution to "small" β

Regularized Least Squares and MAP

What if $(\mathbf{A}^T\mathbf{A})$ is not invertible ?

$$\widehat{\beta}_{\text{MAP}} = \arg\max_{\beta} \underbrace{\log p(\{(X_i, Y_i)\}_{i=1}^n | \beta, \sigma^2)}_{\text{log likelihood}} + \underbrace{\log p(\beta)}_{\text{log prior}}$$

II) Laplace Prior



$$\beta_i \overset{iid}{\sim} \text{Laplace}(0, t) \qquad\qquad p(\beta_i) \propto e^{-|\beta_i|/t}$$

$$\widehat{\beta}_{\text{MAP}} = \arg\min_{\beta} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \lambda \|\beta\|_1$$

$\lambda \downarrow$ constant(σ^2, t)

Lasso

Prior belief that β is Laplace with zero-mean biases solution to “small” β

Ridge Regression vs Lasso

$$\min_{\beta}(\mathbf{A}\beta - \mathbf{Y})^T(\mathbf{A}\beta - \mathbf{Y}) + \lambda \text{pen}(\beta) = \min_{\beta} J(\beta) + \lambda \text{pen}(\beta)$$

Ridge Regression:

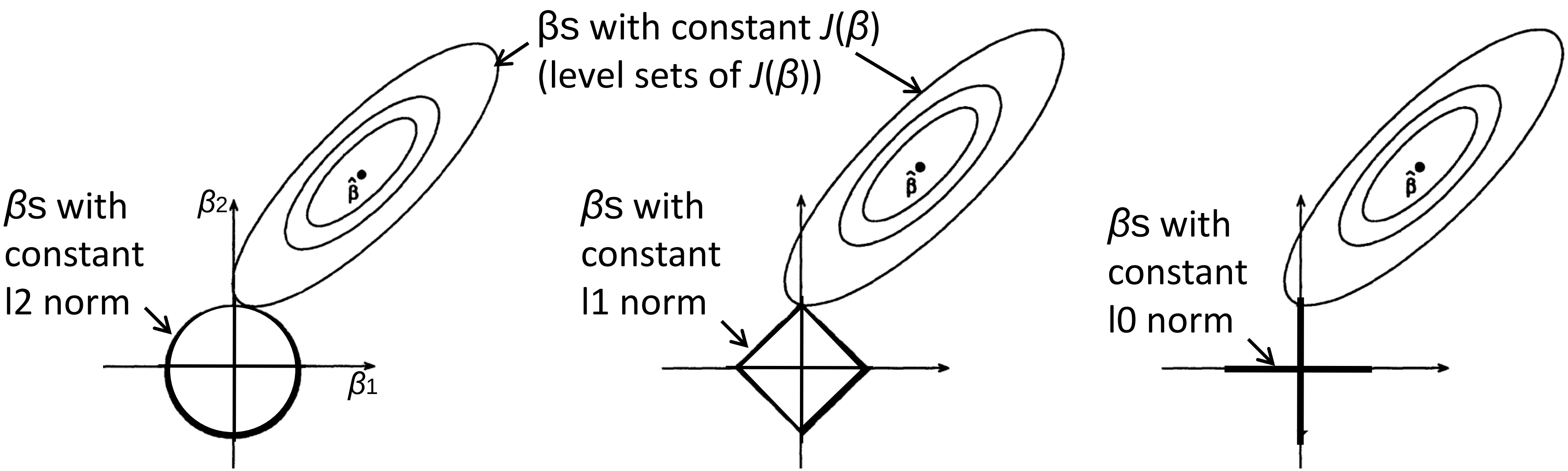
$$\text{pen}(\beta) = \|\beta\|_2^2$$

Lasso:

$$\text{pen}(\beta) = \|\beta\|_1$$

HOT!

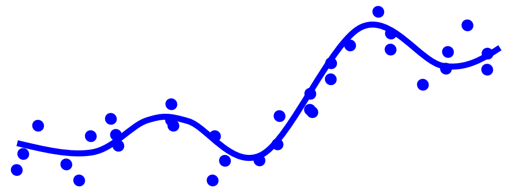
Ideally l0 penalty, but optimization becomes non-convex



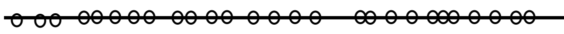
Lasso (l1 penalty) results in sparse solutions – vector with more zero coordinates
Good for high-dimensional problems – don't have to store all coordinates!

Beyond Linear Regression

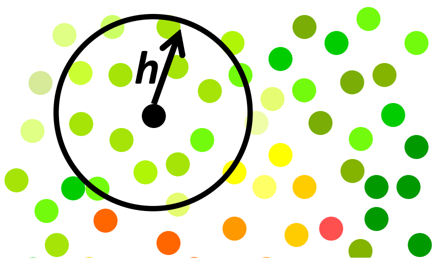
Polynomial regression



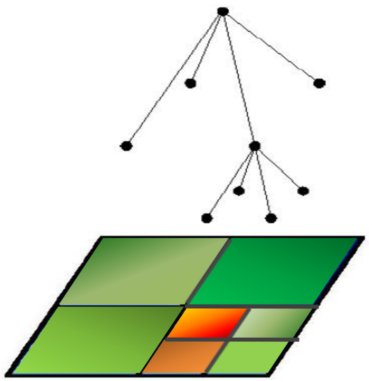
Regression with nonlinear features/basis functions



Kernel regression - Local/Weighted regression



Regression trees – Spatially adaptive regressio



Polynomial Regression

Univariate (1-d) case:

$$f(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_m X^m = \mathbf{X}\beta$$

where $\mathbf{X} = [1 \;\; X \;\; X^2 \ldots X^m], \beta = [\beta_1 \ldots \beta_m]^T$

$$\widehat{\beta} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Y}$$

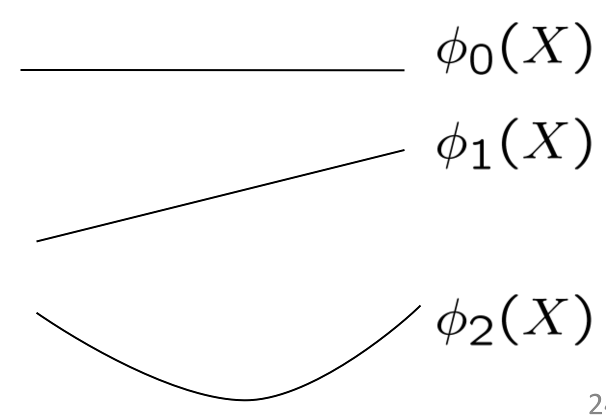
$$\widehat{f}_n(X) = \mathbf{X}\widehat{\beta}$$

$$\mathbf{A} = \begin{bmatrix} 1 & X_1 & X_1^2 & \ldots & X_1^m \\ \vdots & & & \ddots & \vdots \\ 1 & X_n & X_n^2 & \ldots & X_n^m \end{bmatrix}$$

$$f(X) = \sum_{j=0}^{m} \beta_j X^j = \sum_{j=0}^{m} \beta_j \phi_j(X)$$

Weight of each feature

Nonlinear features



Polynomial Regression

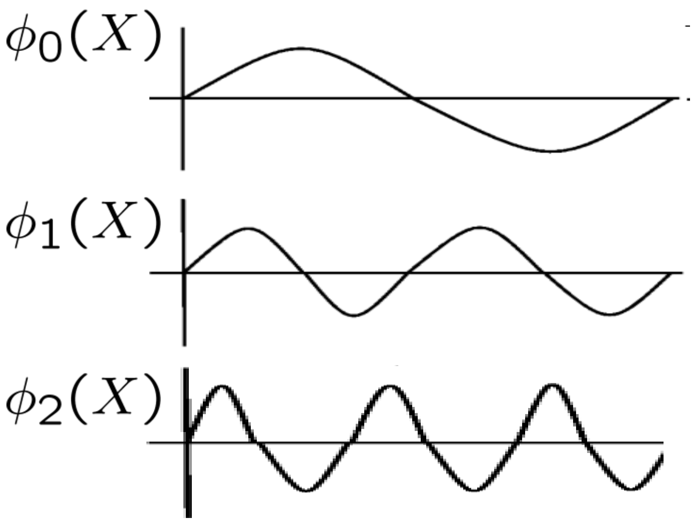
http://mste.illinois.edu/users/exner/java.f/leastsquares/

Nonlinear Regression

$$f(X) = \sum_{j=0}^{m} \beta_j \phi_j(X)$$

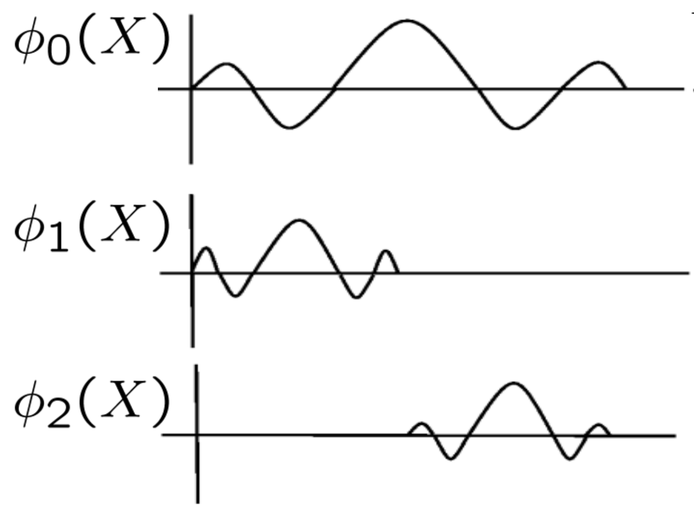
Basis coefficients ← β_j | $\phi_j(X)$: Nonlinear features/basis functions

Fourier Basis



Good representation for oscillatory functions

Wavelet Basis



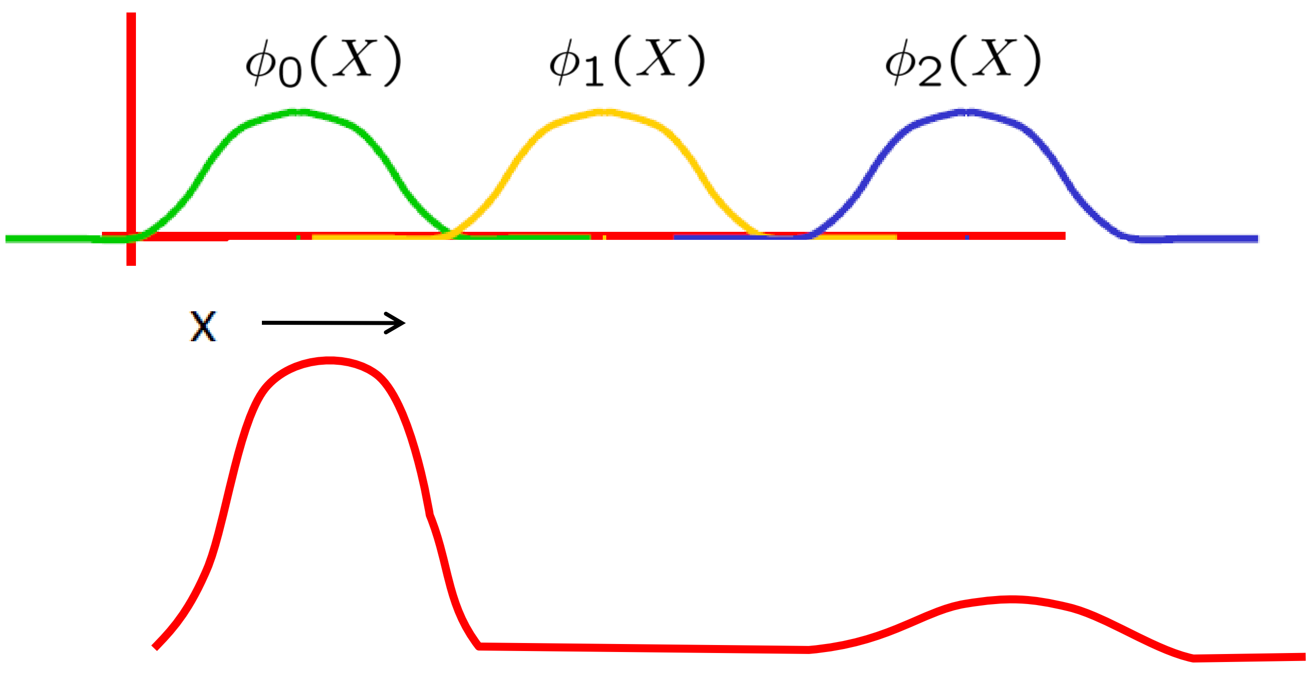
Good representation for functions localized at multiple scales

Local Regression

$$f(X) = \sum_{j=0}^{m} \beta_j \phi_j(X)$$

Basis coefficients ← β_j

Nonlinear features/basis functions ← $\phi_j(X)$

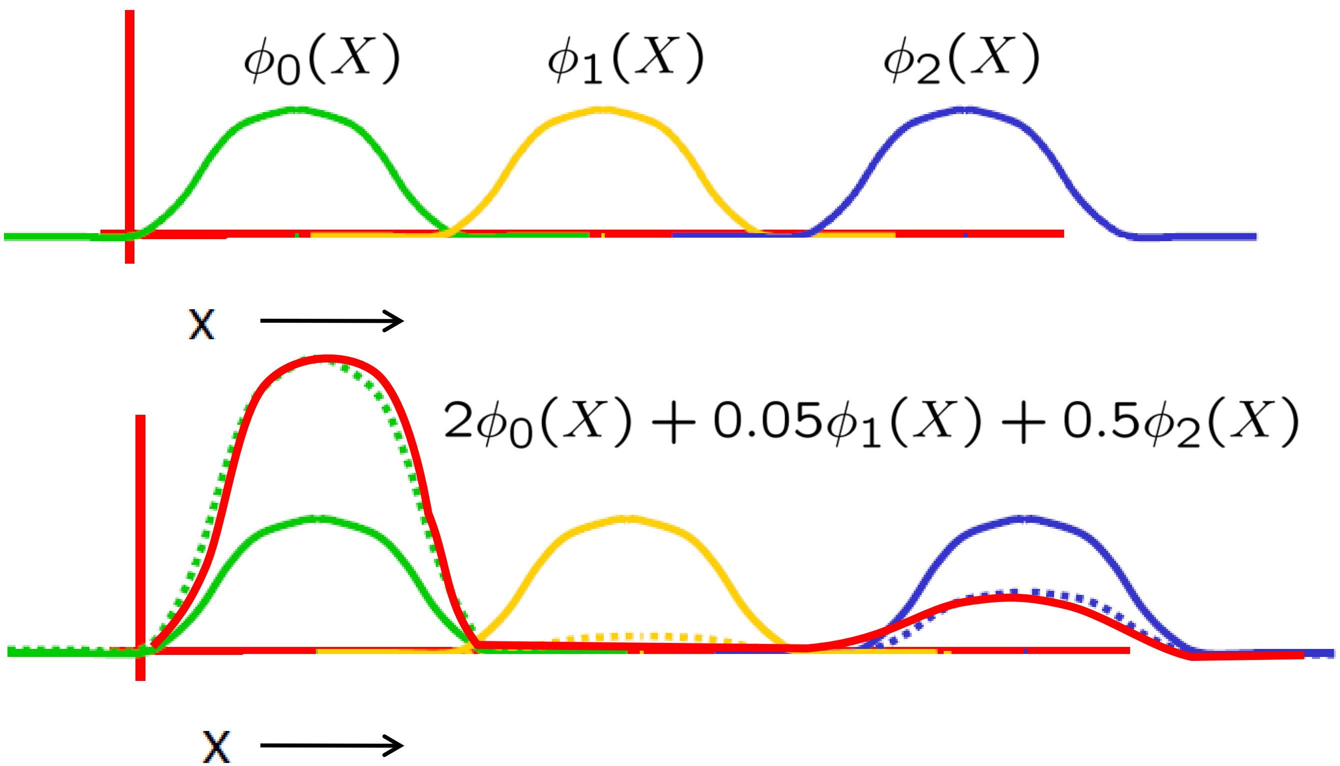


Globally supported basis functions (polynomial, fourier) will not yield a good representation

Local Regression

$$f(X) = \sum_{j=0}^{m} \beta_j \phi_j(X)$$

Basis coefficients ← β_j Nonlinear features/basis functions ← $\phi_j(X)$



Globally supported basis functions (polynomial, fourier) will not yield a good representation

What you should know

Linear Regression

- Least Squares Estimator
- Normal Equations
- Gradient Descent
- Geometric and Probabilistic Interpretation (connection to MLE)

Regularized Linear Regression (connection to MAP)

- Ridge Regression, Lasso

Polynomial Regression, Basis (Fourier, Wavelet) Estimators

Next time

- Kernel Regression (Localized)
- Regression Trees