

Linear Regression contd..

Aarti Singh

Co-instructor: Pradeep Ravikumar

Machine Learning 10-701
Feb 1, 2017



MACHINE LEARNING DEPARTMENT

Carnegie Mellon.
School of Computer Science

Regularized Least Squares

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

r equations , p unknowns – underdetermined system of linear equations
many feasible solutions

Need to constrain solution further

e.g. bias solution to “small” values of β (small changes in input don’t translate to large changes in output)

$$\begin{aligned}\hat{\beta}_{\text{MAP}} &= \arg \min_{\beta} \sum_{i=1}^n (Y_i - X_i \beta)^2 + \lambda \|\beta\|_2^2 && \text{Ridge Regression} \\ &= \arg \min_{\beta} (\mathbf{A}\beta - \mathbf{Y})^T (\mathbf{A}\beta - \mathbf{Y}) + \lambda \|\beta\|_2^2 && \lambda \geq 0\end{aligned}$$

$$\hat{\beta}_{\text{MAP}} = (\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^\top \mathbf{Y}$$

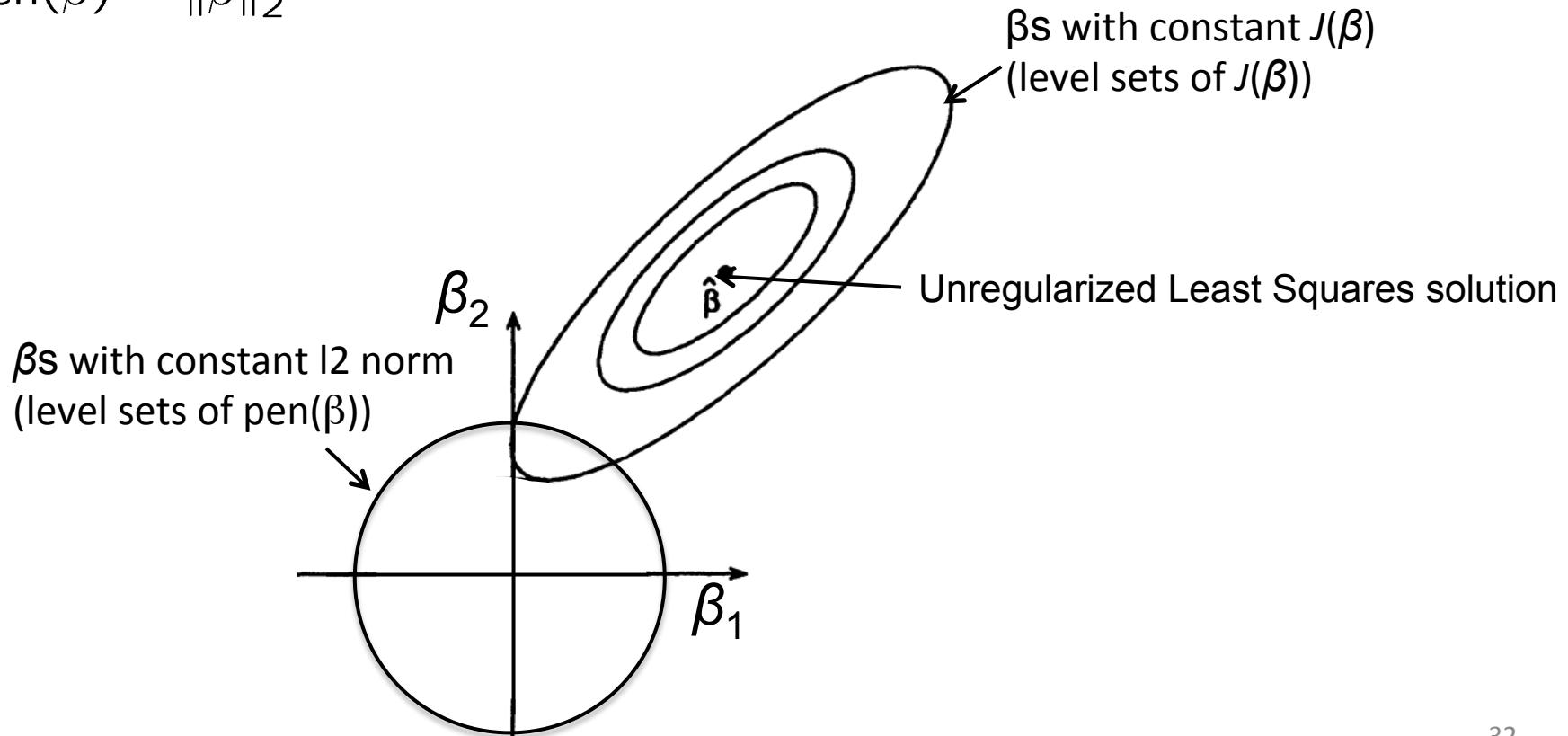
Is $(\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{I})$ invertible ?

Understanding regularized Least Squares

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

Ridge Regression:

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



Regularized Least Squares

What if $(A^T A)$ is not invertible ?

r equations , p unknowns – underdetermined system of linear equations
many feasible solutions

Need to constrain solution further

e.g. bias solution to “small” values of β (small changes in input don’t translate to large changes in output)

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

Ridge Regression
($\|2$ penalty)

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

$\lambda \geq 0$
Lasso
($\|1$ penalty)

Many β can be zero – many inputs are irrelevant to prediction in high-dimensional settings (typically intercept term not penalized)

Regularized Least Squares

What if $(A^T A)$ is not invertible ?

r equations , p unknowns – underdetermined system of linear equations
many feasible solutions

Need to constrain solution further

e.g. bias solution to “small” values of β (small changes in input don’t translate to large changes in output)

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

Ridge Regression
($\|2$ penalty)

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

$\lambda \geq 0$
Lasso
($\|1$ penalty)

No closed form solution, but can optimize using sub-gradient descent (packages available)

Ridge Regression vs Lasso

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

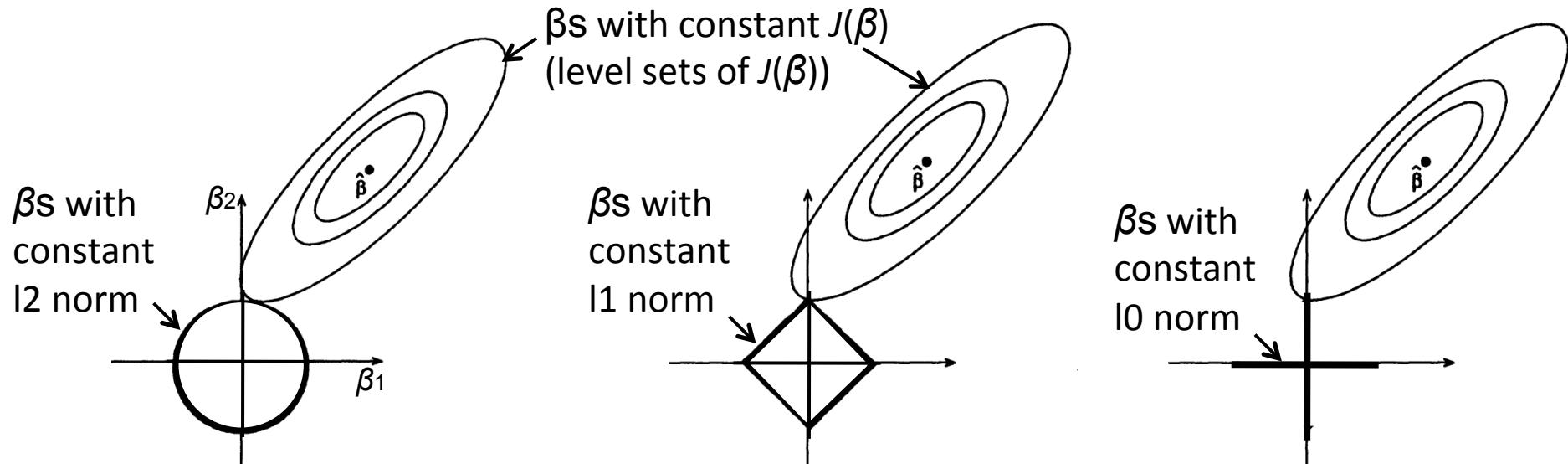
Ridge Regression:

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

Lasso:

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

Ideally ℓ_0 penalty,
but optimization
becomes non-convex



Lasso (ℓ_1 penalty) results in sparse solutions – vector with more zero coordinates
Good for high-dimensional problems – don't have to store all coordinates,
interpretable solution!

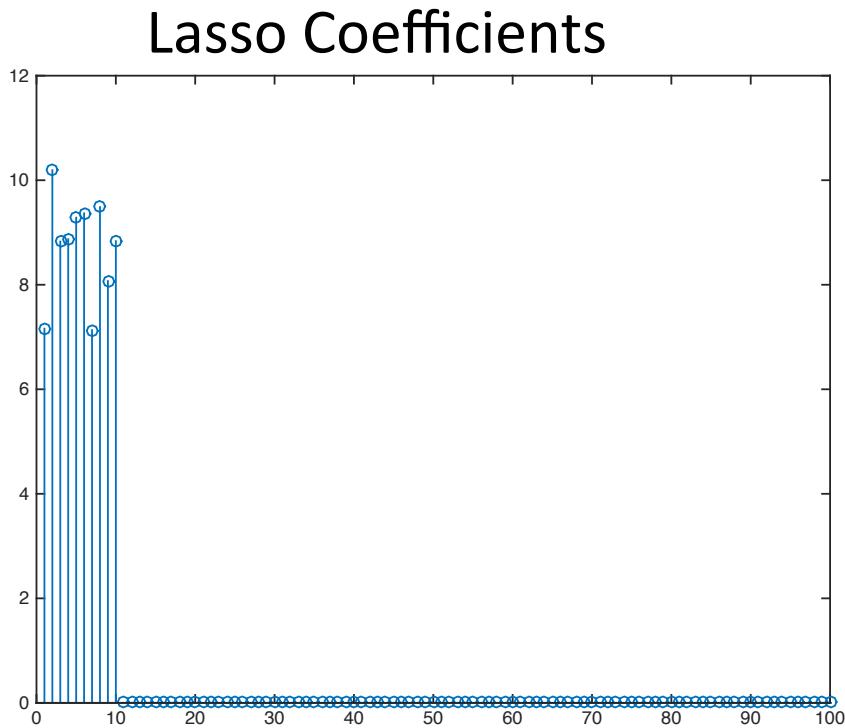
Matlab example

```
clear all  
close all  
  
n = 80;    % datapoints  
p = 100;   % features  
k = 10;    % non-zero features  
  
rng(20);  
X = randn(n,p);  
weights = zeros(p,1);  
weights(1:k) = randn(k,1)+10;  
noise = randn(n,1) * 0.5;  
Y = X*weights + noise;  
  
Xtest = randn(n,p);  
noise = randn(n,1) * 0.5;  
Ytest = Xtest*weights + noise;
```

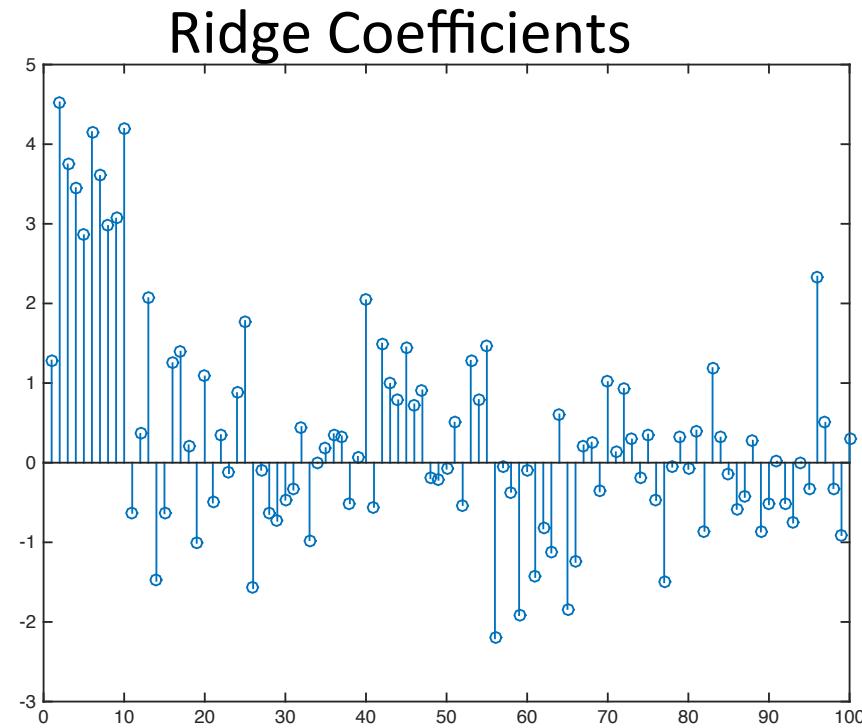
```
lassoWeights = lasso(X,Y,'Lambda',1,  
'Alpha', 1.0);  
Ylasso = Xtest*lassoWeights;  
norm(Ytest-Ylasso)  
  
ridgeWeights = lasso(X,Y,'Lambda',1,  
'Alpha', 0.0001);  
Yridge = Xtest*ridgeWeights;  
norm(Ytest-Yridge)  
  
stem(lassoWeights)  
pause  
stem(ridgeWeights)
```

Matlab example

Test MSE = 33.7997



Test MSE = 185.9948



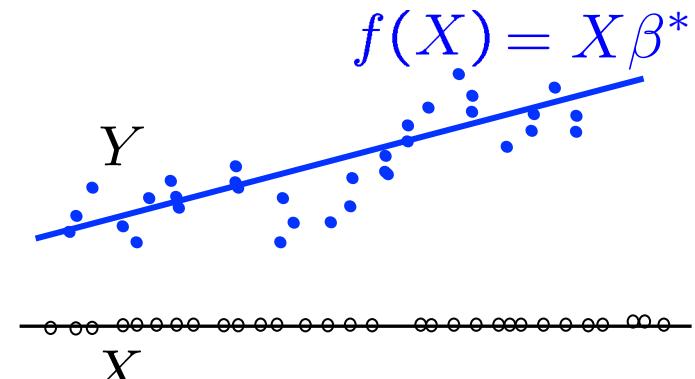
Regularized Least Squares – connection to MLE and MAP (Model-based approaches)

Least Squares and M(C)LE

Intuition: Signal plus (zero-mean) Noise model

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

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



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

Conditional log likelihood

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

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

Regularized Least Squares and M(C)AP

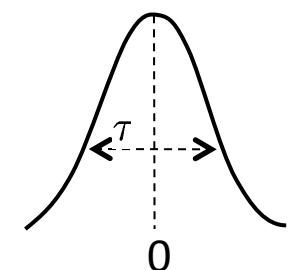
What if $(\mathbf{A}^T \mathbf{A})$ is not invertible ?

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

I) Gaussian Prior

$$\beta \sim \mathcal{N}(0, \tau^2 \mathbf{I})$$

$$p(\beta) \propto e^{-\beta^T \beta / 2\tau^2}$$



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

constant(σ^2, τ^2)

Ridge Regression

$$\hat{\beta}_{\text{MAP}} = (\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^T \mathbf{Y}$$

Regularized Least Squares and M(C)AP

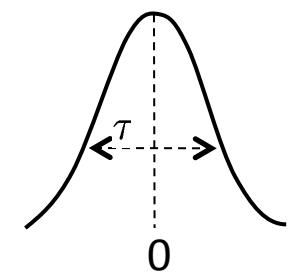
What if $(\mathbf{A}^T \mathbf{A})$ is not invertible ?

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

I) Gaussian Prior

$$\beta \sim \mathcal{N}(0, \tau^2 \mathbf{I})$$

$$p(\beta) \propto e^{-\beta^T \beta / 2\tau^2}$$



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

constant(σ^2, τ^2)

Ridge Regression

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

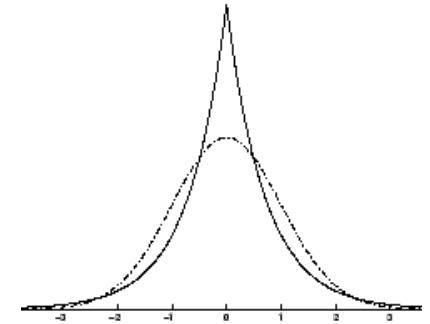
Regularized Least Squares and M(C)AP

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

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

II) Laplace Prior

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



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

↓
constant(σ^2, t)

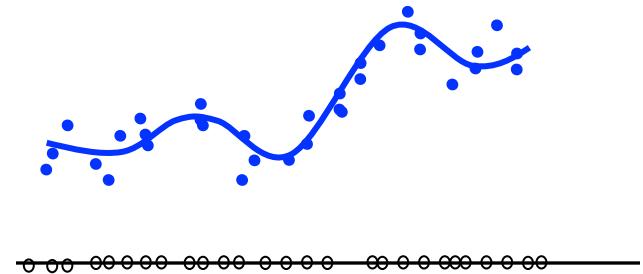
Lasso

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

Beyond Linear Regression

Polynomial regression

Regression with nonlinear features



Kernelized Ridge Regression (later)

Local Kernel Regression (later)

Polynomial Regression

degree m
↓

Univariate (1-dim) $f(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \dots + \beta_m X^m = \mathbf{X}\boldsymbol{\beta}$
case:

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

$$\hat{\boldsymbol{\beta}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{Y} \text{ or } (\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^T \mathbf{Y} \quad \hat{f}_n(X) = \mathbf{X} \hat{\boldsymbol{\beta}}$$

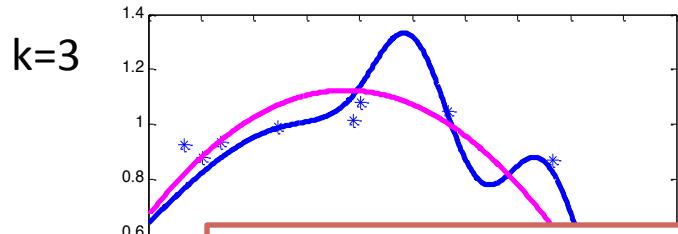
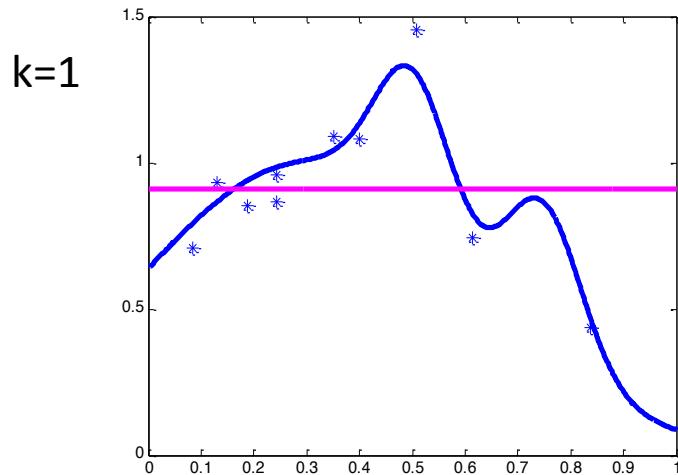
where $\mathbf{A} = \begin{bmatrix} 1 & X_1 & X_1^2 & \dots & X_1^m \\ \vdots & & \ddots & & \vdots \\ 1 & X_n & X_n^2 & \dots & X_n^m \end{bmatrix}$

Multivariate (p-dim) $f(X) = \beta_0 + \beta_1 X^{(1)} + \beta_2 X^{(2)} + \dots + \beta_p X^{(p)}$
case:

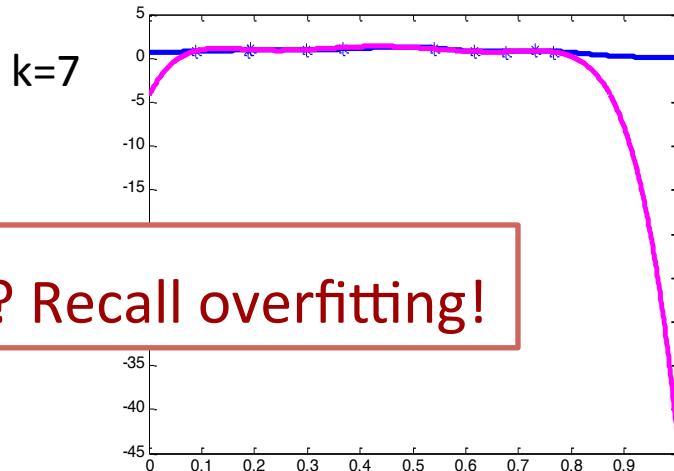
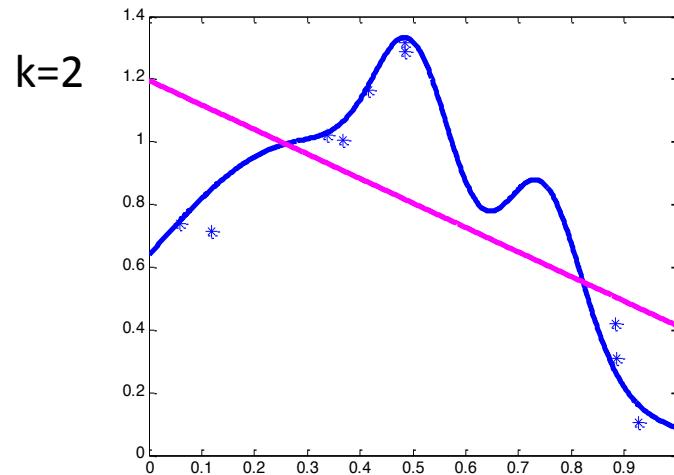
$$\begin{aligned} &+ \sum_{i=1}^p \sum_{j=1}^p \beta_{ij} X^{(i)} X^{(j)} + \sum_{i=1}^p \sum_{j=1}^p \sum_{k=1}^p X^{(i)} X^{(j)} X^{(k)} \\ &+ \dots \text{terms up to degree m} \end{aligned}$$

Polynomial Regression

Polynomial of order k , equivalently of degree up to $k-1$



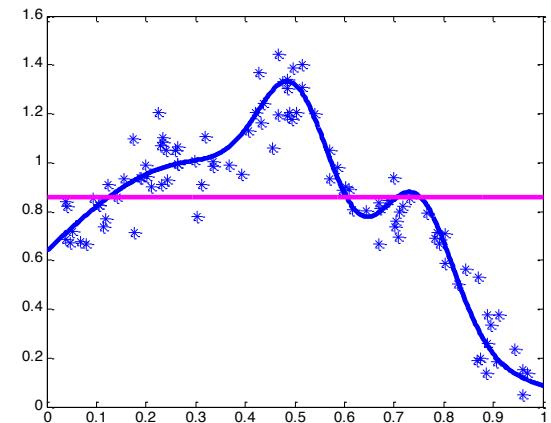
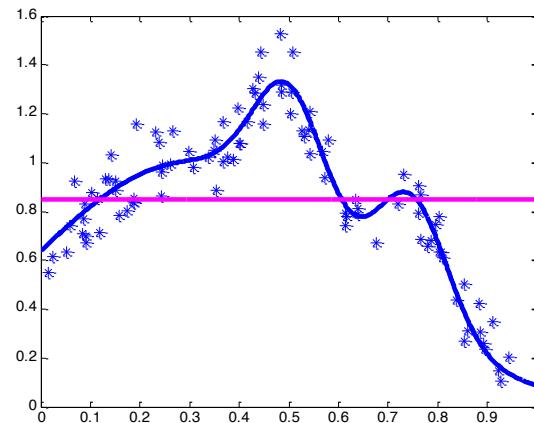
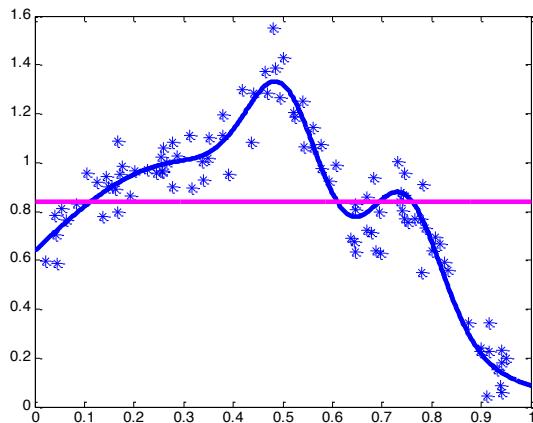
What is the right order? Recall overfitting!



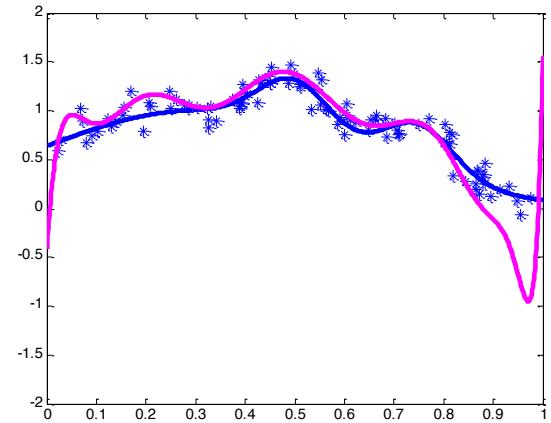
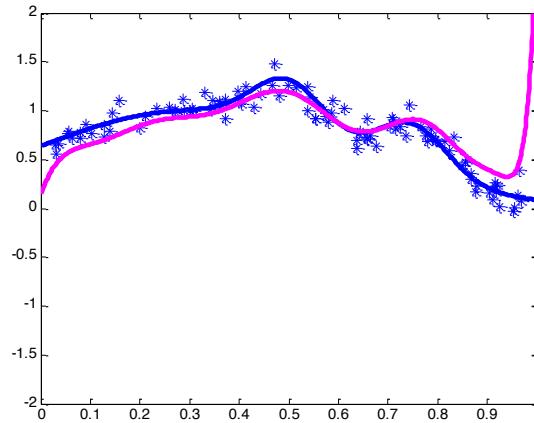
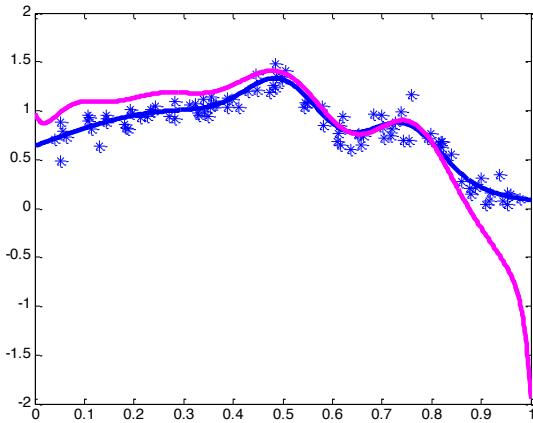
Bias – Variance Tradeoff

3 Independent training datasets

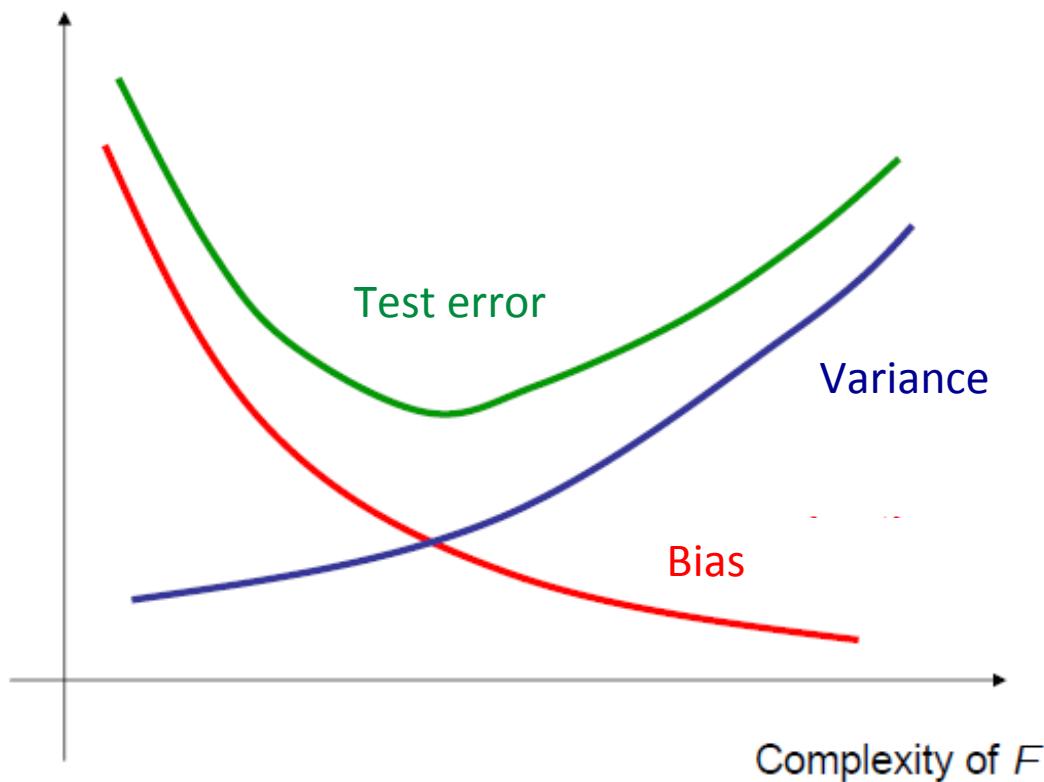
Large bias, Small variance – poor approximation but robust/stable



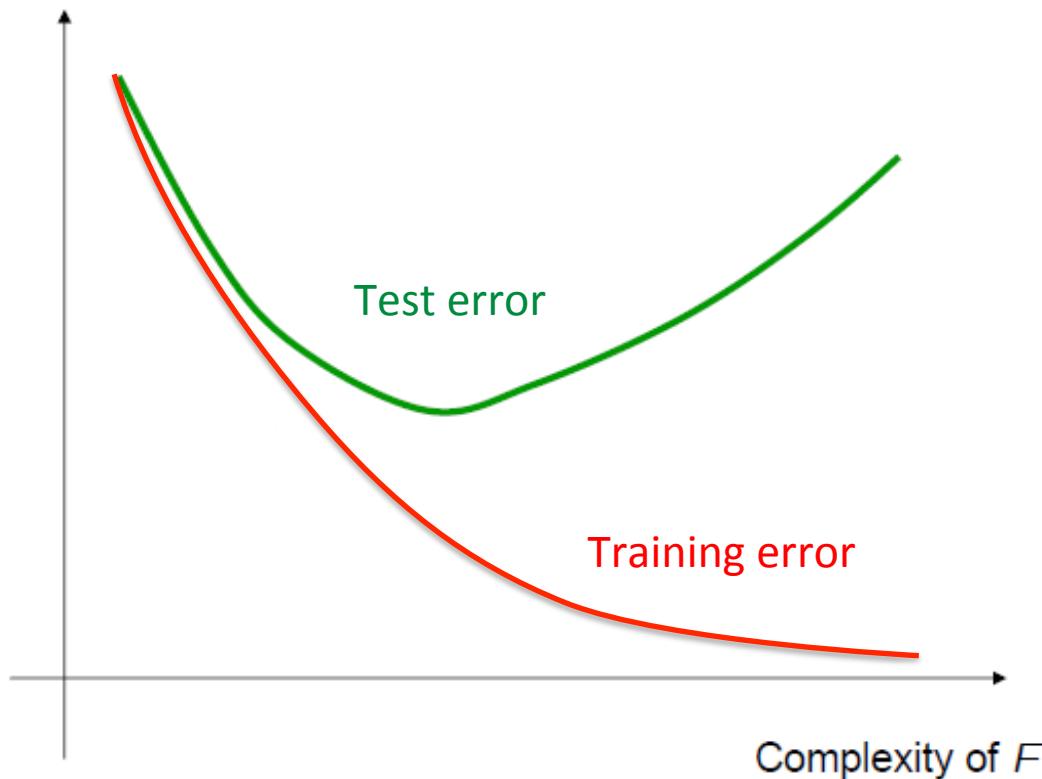
Small bias, Large variance – good approximation but unstable



Effect of Model Complexity



Effect of Model Complexity



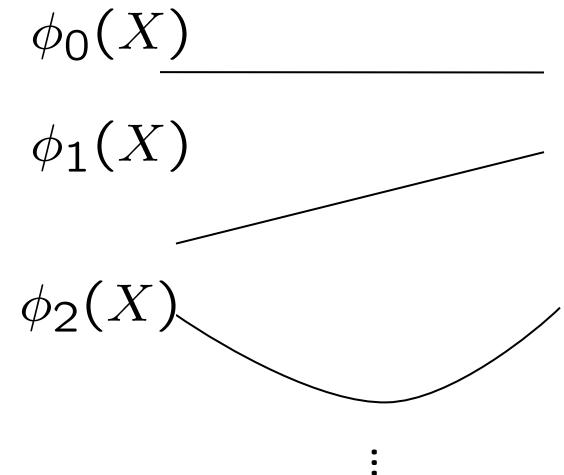
Regression with basis functions

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

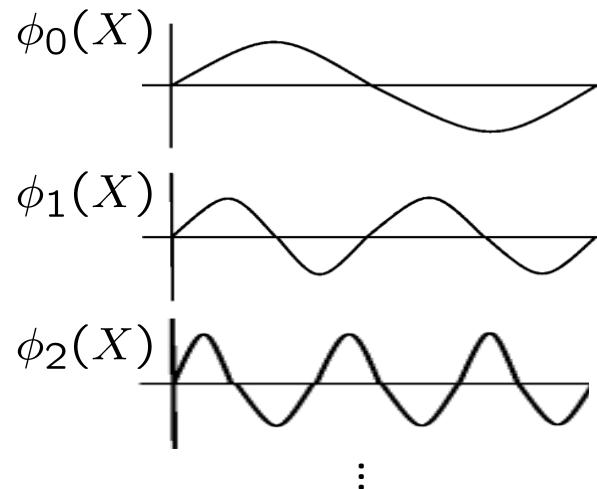
Basis coefficients

Basis functions (Linear combinations yield meaningful spaces)

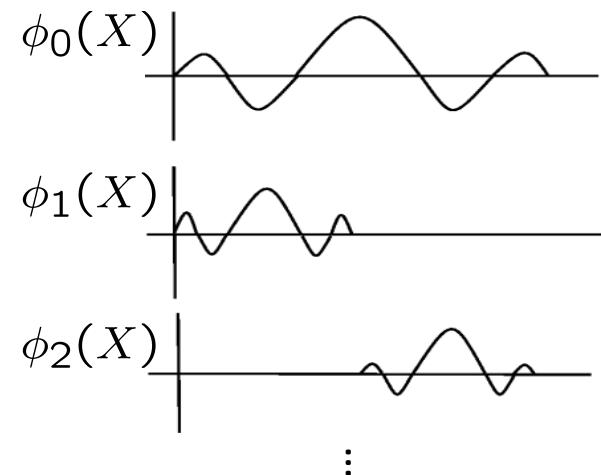
Polynomial Basis



Fourier Basis



Wavelet Basis



Good representation for
periodic functions

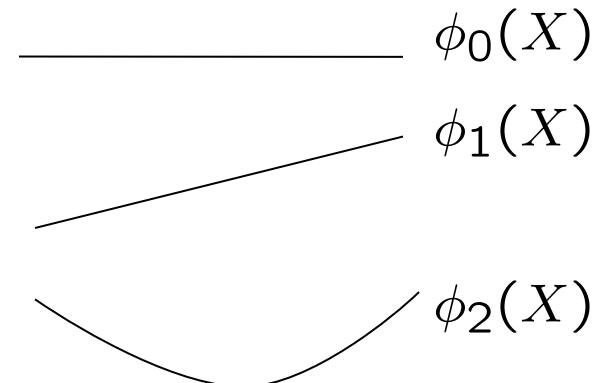
Good representation for
local functions

Regression with nonlinear features

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

Weight of
each feature

Nonlinear
features



In general, use any nonlinear features

e.g. e^X , $\log X$, $1/X$, $\sin(X)$, ...

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

or

$$(\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^T \mathbf{Y}$$

$$\mathbf{A} = \begin{bmatrix} \phi_0(X_1) & \phi_1(X_1) & \dots & \phi_m(X_1) \\ \vdots & \ddots & & \vdots \\ \phi_0(X_n) & \phi_1(X_n) & \dots & \phi_m(X_n) \end{bmatrix}$$

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

$$\mathbf{X} = [\phi_0(X) \ \phi_1(X) \ \dots \ \phi_m(X)]$$

Logistic Regression

Aarti Singh

Co-instructor: Pradeep Ravikumar

Machine Learning 10-701
Feb 1, 2017

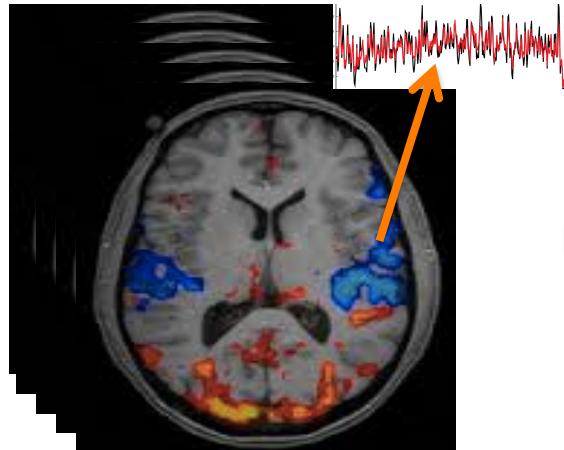


MACHINE LEARNING DEPARTMENT

Carnegie Mellon.
School of Computer Science

Regression to Classification

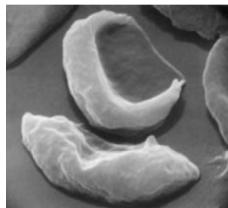
Regression



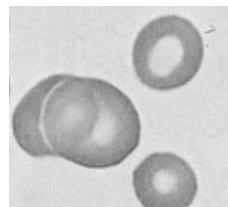
$Y = \text{Age of a subject}$

$X = \text{Brain Scan}$

Classification



Anemic cell
Healthy cell



$X = \text{Cell Image}$

$Y = \text{Diagnosis}$

Can we predict the “probability” of class label being Anemic or Healthy – a real number using regression methods?

But output (probability) needs to be $[0,1]$

Logistic Regression

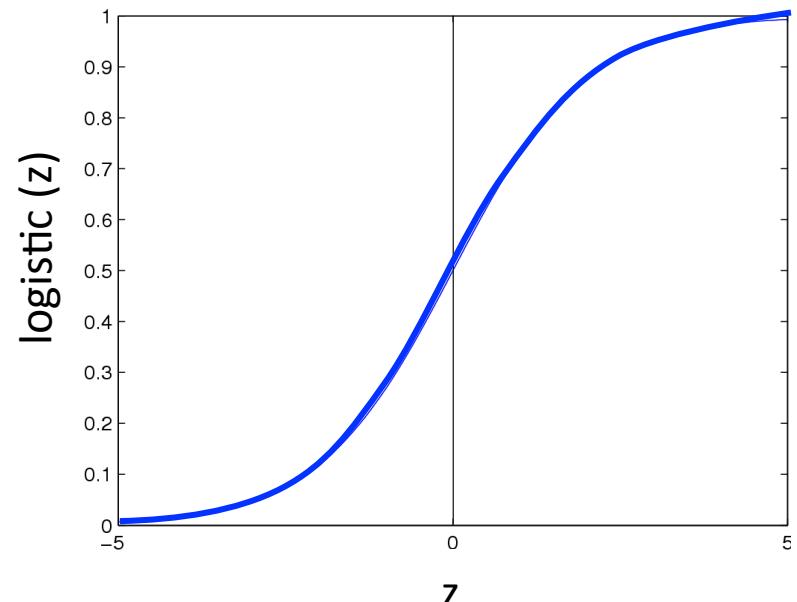
Not really regression

Assumes the following functional form for $P(Y|X)$:

$$P(Y = 0|X) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

Logistic function applied to a linear function of the data

Logistic function
(or Sigmoid): $\frac{1}{1 + \exp(-z)}$



Features can be discrete or continuous!

Logistic Regression is a Linear Classifier!

Assumes the following functional form for $P(Y|X)$:

$$P(Y = 0|X) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

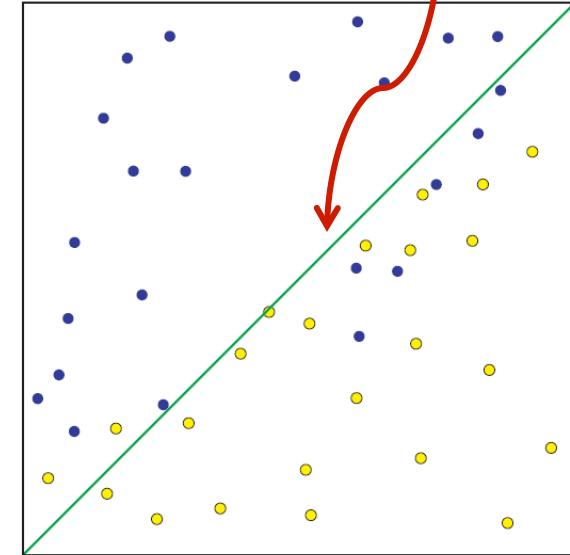
$$w_0 + \sum_i w_i X_i = 0$$

Decision boundary: Note - Labels are 0,1

$$P(Y = 0|X) \stackrel{0}{\underset{1}{\gtrless}} P(Y = 1|X)$$

$$w_0 + \sum_i w_i X_i \stackrel{1}{\underset{0}{\gtrless}} 0$$

(Linear Decision Boundary)



Logistic Regression is a Linear Classifier!

Assumes the following functional form for $P(Y|X)$:

$$P(Y = 0|X) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

$$\Rightarrow P(Y = 1|X) = \frac{\exp(w_0 + \sum_i w_i X_i)}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

$$\Rightarrow \frac{P(Y = 1|X)}{P(Y = 0|X)} = \exp(w_0 + \sum_i w_i X_i) \stackrel{1}{\underset{0}{\gtrless}} 1$$

$$\Rightarrow w_0 + \sum_i w_i X_i \stackrel{1}{\underset{0}{\gtrless}} 0$$

Training Logistic Regression

How to learn the parameters w_0, w_1, \dots, w_d ? (d features)

Training Data $\{(X^{(j)}, Y^{(j)})\}_{j=1}^n$ $X^{(j)} = (X_1^{(j)}, \dots, X_d^{(j)})$

Maximum Likelihood Estimates

$$\hat{\mathbf{w}}_{MLE} = \arg \max_{\mathbf{w}} \prod_{j=1}^n P(X^{(j)}, Y^{(j)} | \mathbf{w})$$

But there is a problem ...

Don't have a model for $P(X)$ or $P(X|Y)$ – only for $P(Y|X)$

Training Logistic Regression

How to learn the parameters w_0, w_1, \dots, w_d ? (d features)

Training Data $\{(X^{(j)}, Y^{(j)})\}_{j=1}^n$ $X^{(j)} = (X_1^{(j)}, \dots, X_d^{(j)})$

Maximum (Conditional) Likelihood Estimates

$$\hat{\mathbf{w}}_{MCLE} = \arg \max_{\mathbf{w}} \prod_{j=1}^n P(Y^{(j)} | X^{(j)}, \mathbf{w})$$

Discriminative philosophy – Don't waste effort learning $P(X)$,
focus on $P(Y|X)$ – that's all that matters for classification!

Expressing Conditional log Likelihood

$$P(Y = 0 | \mathbf{X}, \mathbf{w}) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

$$P(Y = 1 | \mathbf{X}, \mathbf{w}) = \frac{\exp(w_0 + \sum_i w_i X_i)}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

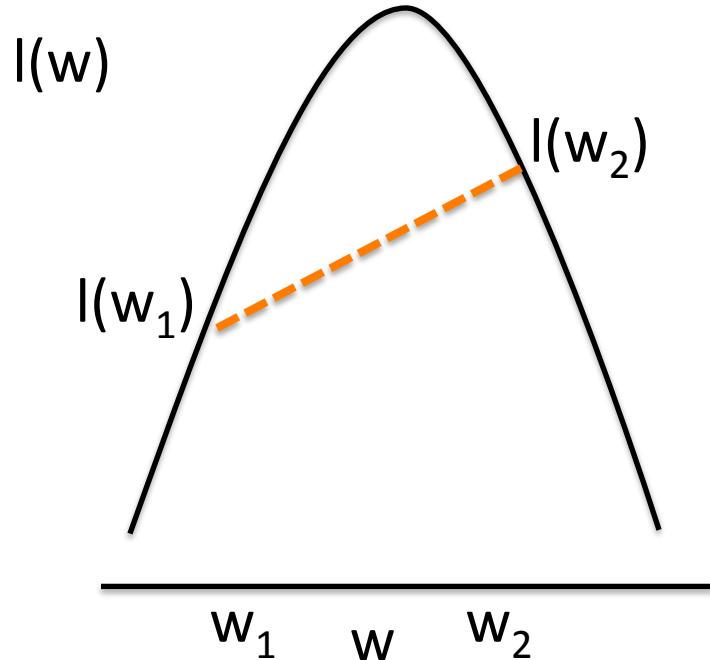
$$\begin{aligned} l(\mathbf{w}) &\equiv \ln \prod_j P(y^j | \mathbf{x}^j, \mathbf{w}) \\ &= \sum_j \left[y^j (w_0 + \sum_i^d w_i x_i^j) - \ln(1 + \exp(w_0 + \sum_i^d w_i x_i^j)) \right] \end{aligned}$$

Bad news: no closed-form solution to maximize $l(\mathbf{w})$

Good news: $l(\mathbf{w})$ is concave function of \mathbf{w} !

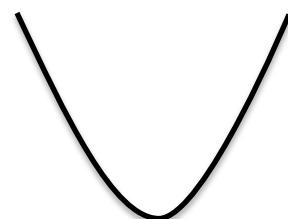
concave functions easy to maximize (unique maximum)

Concave function

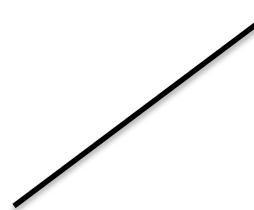


A function $l(w)$ is called **concave** if the line joining two points $l(w_1), l(w_2)$ on the function does not go above the function on the interval $[w_1, w_2]$

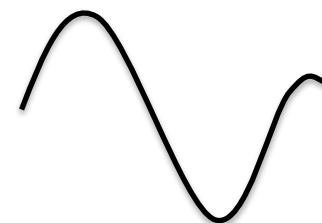
(Strictly) Concave functions have a unique maximum!



Convex



Both Concave & Convex

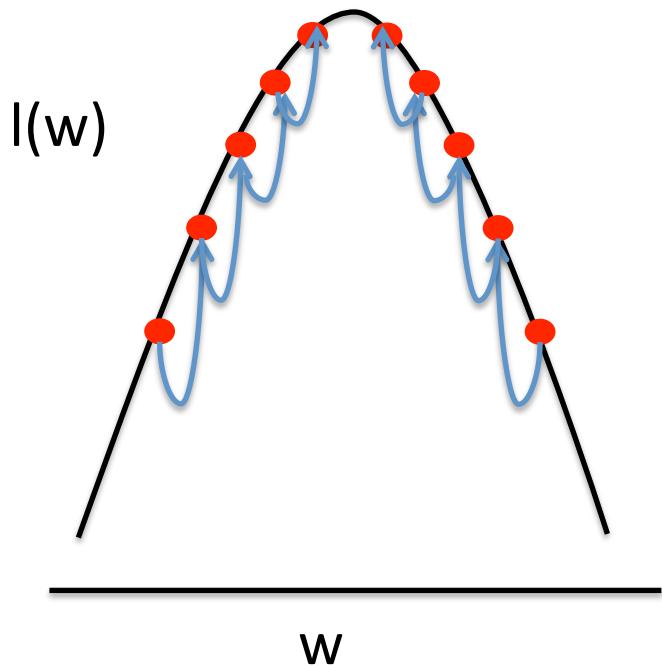


Neither

Optimizing concave function

- Conditional likelihood for Logistic Regression is concave
- Maximum of a concave function can be reached by

Gradient Ascent Algorithm



Initialize: Pick \mathbf{w} at random

Gradient:

$$\nabla_{\mathbf{w}} l(\mathbf{w}) = \left[\frac{\partial l(\mathbf{w})}{\partial w_0}, \dots, \frac{\partial l(\mathbf{w})}{\partial w_d} \right]'$$

Update rule: Learning rate, $\eta > 0$

$$\Delta \mathbf{w} = \eta \nabla_{\mathbf{w}} l(\mathbf{w})$$

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \frac{\partial l(\mathbf{w})}{\partial w_i} \Big|_t$$

Gradient Ascent for Logistic Regression

Gradient ascent rule for w_0 :

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \frac{\partial l(\mathbf{w})}{\partial w_0} \Big|_t$$

$$l(\mathbf{w}) = \sum_j \left[y^j (w_0 + \sum_i^d w_i x_i^j) - \ln(1 + \exp(w_0 + \sum_i^d w_i x_i^j)) \right]$$

$$\frac{\partial l(\mathbf{w})}{\partial w_0} = \sum_j \left[y^j - \underbrace{\frac{1}{1 + \exp(w_0 + \sum_i^d w_i x_i^j)} \cdot \exp(w_0 + \sum_i^d w_i x_i^j)}_{\text{gradient term}} \right]$$

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})]$$

Gradient Ascent for Logistic Regression

Gradient ascent algorithm: iterate until change < ε

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})]$$

For $i=1, \dots, d$,

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})]$$

repeat

Predict what current weight thinks label Y should be

- Gradient ascent is simplest of optimization approaches
 - e.g., Newton method, Conjugate gradient ascent, IRLS (see Bishop 4.3.3)

That's all M(C)LE. How about M(C)AP?

$$p(\mathbf{w} \mid Y, \mathbf{X}) \propto P(Y \mid \mathbf{X}, \mathbf{w}) p(\mathbf{w})$$

- Define priors on \mathbf{w}
 - Common assumption: Normal distribution, zero mean, identity covariance
 - “Pushes” parameters towards zero

$$p(\mathbf{w}) = \prod_i \frac{1}{\kappa \sqrt{2\pi}} e^{\frac{-w_i^2}{2\kappa^2}}$$

Zero-mean Gaussian prior

- M(C)AP estimate $\mathbf{w}^* = \arg \max_{\mathbf{w}} \ln \left[p(\mathbf{w}) \prod_{j=1}^n P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$

$$\mathbf{w}^* = \arg \max_{\mathbf{w}} \sum_{j=1}^n \ln P(y^j \mid \mathbf{x}^j, \mathbf{w}) - \underbrace{\sum_{i=1}^d \frac{w_i^2}{2\kappa^2}}$$

Still concave objective!

Penalizes large weights

M(C)AP – Gradient

- Gradient

$$p(\mathbf{w}) = \prod_i \frac{1}{\kappa \sqrt{2\pi}} e^{\frac{-w_i^2}{2\kappa^2}}$$

$$\frac{\partial}{\partial w_i} \ln \left[p(\mathbf{w}) \prod_{j=1}^n P(y^j | \mathbf{x}^j, \mathbf{w}) \right]$$

Zero-mean Gaussian prior

$$\frac{\partial}{\partial w_i} \ln p(\mathbf{w}) + \frac{\partial}{\partial w_i} \ln \left[\prod_{j=1}^n P(y^j | \mathbf{x}^j, \mathbf{w}) \right]$$

Same as before

$$\rightarrow \propto \frac{-w_i}{\kappa^2}$$

Extra term Penalizes large weights

M(C)LE vs. M(C)AP

- Maximum conditional likelihood estimate

$$\mathbf{w}^* = \arg \max_{\mathbf{w}} \ln \left[\prod_{j=1}^n P(y^j | \mathbf{x}^j, \mathbf{w}) \right]$$

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - P(Y = 1 | \mathbf{x}^j, \mathbf{w}^{(t)})]$$

- Maximum conditional a posteriori estimate

$$\mathbf{w}^* = \arg \max_{\mathbf{w}} \ln \left[p(\mathbf{w}) \prod_{j=1}^n P(y^j | \mathbf{x}^j, \mathbf{w}) \right]$$

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \left\{ -\frac{1}{\kappa^2} w_i^{(t)} + \sum_j x_i^j [y^j - P(Y = 1 | \mathbf{x}^j, \mathbf{w}^{(t)})] \right\}$$

Logistic Regression for more than 2 classes

- Logistic regression in more general case, where $Y \in \{y_1, \dots, y_K\}$

for $k < K$

$$P(Y = y_k | X) = \frac{\exp(w_{k0} + \sum_{i=1}^d w_{ki} X_i)}{1 + \sum_{j=1}^{K-1} \exp(w_{j0} + \sum_{i=1}^d w_{ji} X_i)}$$

for $k = K$ (normalization, so no weights for this class)

$$P(Y = y_K | X) = \frac{1}{1 + \sum_{j=1}^{K-1} \exp(w_{j0} + \sum_{i=1}^d w_{ji} X_i)}$$

Predict $f^*(x) = \arg \max_{Y=y} P(Y = y | X = x)$

Is the decision boundary still linear?