Discrete to Continuous Labels

Classification

X = Document  Y = Topic  X = Cell Image  Y = Diagnosis
Sports  Science  Anemic cell  Healthy cell
News

Regression

Stock Market Prediction

X = Feb01  Y = ?
Regression Tasks

Weather Prediction

Y = Temp
X = 7 pm
Supervised Learning

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

**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:

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

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

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: \[ R(f) \geq R(f^*) \text{ for any prediction rule } f \]

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

Dropping subscripts for notational convenience

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

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

\[ = \mathbb{E} \left[ (f(X) - E[Y|X])^2|X \right] + 2(f(X) - E[Y|X]) \times 0 + \mathbb{E} \left[ (E[Y|X] - Y)^2|X \right] \]

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

Optimal predictor:

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

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

Intuition: Signal plus (zero-mean) Noise model

\[ Y = f^*(X) + \epsilon \]

Depends on unknown distribution \( P_{XY} \)
Regression algorithms

Training data \[ \{(X_i, Y_i)\}_{i=1}^n \] → Learning algorithm → Prediction rule \[ \hat{f}_n \]

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:

\[ \hat{f}_n = \arg \min_{f \in \mathcal{F}} \left( \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2 \right) \]

Class of predictors \hspace{1cm} \text{Empirical mean}

\[
\frac{1}{n} \sum_{i=1}^{n} [\text{loss}(Y_i, f(X_i))] \xrightarrow{\text{Law of Large Numbers}} \mathbb{E}_{XY}[\text{loss}(Y, f(X))]
\]
ERM – you saw it before!

• Learning Distributions

Max likelihood = Min -ve log likelihood empirical risk

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

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

Class of parametric distributions

Bernoulli ($\theta$)
Gaussian ($\mu, \sigma^2$)
Linear Regression

\[ \hat{f}_n^L = \arg \min_{f \in \mathcal{F}_L} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2 \quad \text{Least Squares Estimator} \]

\( \mathcal{F}_L \) - Class of Linear functions

Uni-variate case:

\[ f(X) = \beta_1 + \beta_2 X \]

\( \beta_1 \) - intercept

Multi-variate case:

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

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

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

\[ \hat{\beta} = \arg \min_{\beta} \frac{1}{n} \sum_{i=1}^{n} (X_i \beta - Y_i)^2 \]
\[ \hat{f}_n^L(X) = X \hat{\beta} \]

\[ \arg \min_{\beta} \frac{1}{n} (A \beta - Y)^T (A \beta - Y) \]

\[ A = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} X_1^{(1)} & \cdots & X_1^{(p)} \\ \vdots & \ddots & \vdots \\ X_n^{(1)} & \cdots & X_n^{(p)} \end{bmatrix} \]
\[ Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} \]
Least Squares Estimator

\[ \hat{\beta} = \arg \min_{\beta} \frac{1}{n} (A\beta - Y)^T (A\beta - Y) = \arg \min_{\beta} J(\beta) \]

\[ J(\beta) = (A\beta - Y)^T (A\beta - Y) = A^T A\beta \beta^T - 2\beta^T A^T Y + Y^T Y \]

\[ \frac{\partial J(\beta)}{\partial \beta} \bigg|_{\hat{\beta}} = 0 \quad \Rightarrow \quad 2A^T A\beta - 2A^T Y = 0 \]
Normal Equations

\[(A^TA)\hat{\beta} = A^TY\]

If \((A^TA)\) is invertible,

\[\hat{\beta} = (A^TA)^{-1}A^TY\]

When is \((A^TA)\) invertible?
Recall: Full rank matrices are invertible. What is rank of \((A^TA)\)?

What if \((A^TA)\) is not invertible?
Regularization (later)
Revisiting Gradient Descent

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

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

Gradient Descent since $J(\beta)$ is convex

Initialize: $\beta^0$

Update: $\beta^{t+1} = \beta^t - \frac{\alpha}{2} \frac{\partial J(\beta)}{\partial \beta} \bigg|_t$

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

0 if $\beta^t = \hat{\beta}$

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

Effect of step-size $\alpha$

Large $\alpha$ => Fast convergence but larger residual error
Also possible oscillations

Small $\alpha$ => Slow convergence but small residual error
Least Squares and MLE

Intuition: Signal plus (zero-mean) Noise model

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

\[ \epsilon \sim \mathcal{N}(0, \sigma^2 I) \]

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

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

\[ \log \text{likelihood} \]

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

Least Square Estimate is same as Maximum Likelihood Estimate under a Gaussian model!
An early demonstration of the strength of Gauss's method came when it was used to predict the future location of the newly discovered asteroid Ceres. On January 1, 1801, the Italian astronomer Giuseppe Piazzi discovered Ceres and was able to track its path for 40 days before it was lost in the glare of the sun. Based on this data, astronomers desired to determine the location of Ceres after it emerged from behind the sun without solving the complicated Kepler's nonlinear equations of planetary motion. The only predictions that successfully allowed Hungarian astronomer Franz Xaver von Zach to relocate Ceres were those performed by the 24-year-old Gauss using least-squares analysis.

Regularized Least Squares and MAP

What if \((A^TA)\) is not invertible?

\[
\hat{\beta}_{\text{MAP}} = \arg \max \beta \log p((X_i, Y_i)_{i=1}^n | \beta, \sigma^2) + \log p(\beta)
\]

log likelihood log prior

1) Gaussian Prior

\[
\beta \sim N(0, \tau^2 I) \quad p(\beta) \propto e^{-\beta^T \beta / 2\tau^2}
\]

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

constant\((\sigma^2, \tau^2)\)

Ridge Regression

Prior belief that \(\beta\) is Gaussian with zero-mean biases solution to “small” \(\beta\)
Regularized Least Squares and MAP

What if \((A^TA)\) is not invertible?

\[
\hat{\beta}_{\text{MAP}} = \arg \max_{\beta} \log p(\{(X_i, Y_i)\}_{i=1}^{n} | \beta, \sigma^2) + \log p(\beta)
\]

log likelihood \hspace{1cm} \text{log prior}

II) Laplace Prior

\(\beta_i \sim_{\text{iid}} \text{Laplace}(0, t)\)

\(p(\beta_i) \propto e^{-|\beta_i|/t}\)

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

\(\downarrow\) constant\((\sigma^2, t)\)

Prior belief that \(\beta\) is Laplace with zero-mean biases solution to “small” \(\beta\)
Ridge Regression vs Lasso

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

Ridge Regression:
\[
\text{pen}(\beta) = \|\beta\|_2^2
\]

Lasso:
\[
\text{pen}(\beta) = \|\beta\|_1
\]

\beta_1 \quad \beta_2

\beta_1 \quad \beta_2

\beta_1 \quad \beta_2

\beta_1 \quad \beta_2

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 regression
Polynomial Regression

Univariate (1-d) case:

\[ f(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_m X^m = X\beta \]

where \( X = [1 \ X \ X^2 \ldots X^m] \), \( \beta = [\beta_1 \ldots \beta_m]^T \)

\[
\hat{\beta} = (A^T A)^{-1} A^T Y \\
\hat{f}_n(X) = X\hat{\beta}
\]

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

Weight of each feature Nonlinear features

\[ \phi_0(X) \]
\[ \phi_1(X) \]
\[ \phi_2(X) \]
A Regression Example

Average height and weight of American women aged 30 - 39

<table>
<thead>
<tr>
<th>Height/m</th>
<th>1.47</th>
<th>1.5</th>
<th>1.52</th>
<th>1.55</th>
<th>1.57</th>
<th>1.60</th>
<th>1.63</th>
<th>1.65</th>
<th>1.68</th>
<th>1.7</th>
<th>1.73</th>
<th>1.75</th>
<th>1.78</th>
<th>1.8</th>
<th>1.83</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight/kg</td>
<td>52.21</td>
<td>53.12</td>
<td>54.48</td>
<td>55.84</td>
<td>57.2</td>
<td>58.57</td>
<td>59.93</td>
<td>61.29</td>
<td>63.11</td>
<td>64.47</td>
<td>66.28</td>
<td>68.1</td>
<td>69.92</td>
<td>72.19</td>
<td>74.46</td>
</tr>
</tbody>
</table>

Weight is not linear with height, so add a quadratic term into regression

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$$

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

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

$$\hat{\beta}_0 = ?$$

$$\hat{\beta}_1 = ?$$

$$\hat{\beta}_2 = ?$$

Assignment 3 – Programming 1

• Write programs in Matlab, R, C/C++, Java, Perl, or Python to implement the analytical (e.g. matrix-based) or iterative (e.g. gradient descent) linear regression algorithm and test it on the problem in the previous slide. Don’t directly call linear regression functions in any software.

• Turn in the programs and execution results
Assignment 3 – Programming 2
Due Sept. 28, 2015

• Write a program to implement the iterative (e.g. gradient ascent / descent) logistic regression algorithm for binary classification and apply it to the Iris classification data set

• Iris data set: http://archive.ics.uci.edu/ml/datasets/Iris

• Only select data points of two highlighted classes (Iris Setosa, Iris Versicolour, Iris Virginica)

• Submit programs and execution results
Nonlinear Regression

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

Basis coefficients \leftrightarrow Nonlinear features/basis functions

**Fourier Basis**

\[ \phi_0(X) \]

\[ \phi_1(X) \]

\[ \phi_2(X) \]

Good representation for oscillatory functions

\[
\frac{1}{\sqrt{2\pi a^2}} \cdot \text{sinc} \left( \frac{\omega}{2\pi a} \right)
\]

**Wavelet Basis**

\[ \phi_0(X) \]

\[ \phi_1(X) \]

\[ \phi_2(X) \]

Good representation for functions localized at multiple scales

\[
\psi(t) = 2 \text{sinc}(2t) - \text{sinc}(t) = \frac{\sin(2\pi t) - \sin(\pi t)}{\pi t}
\]
Local Regression

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

Basis coefficients \( \beta_j \)
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 → Nonlinear features/basis functions

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

Regularized Linear Regression (connection to MAP)
  Ridge Regression, Lasso

Polynomial Regression, Basis (Fourier, Wavelet) Estimators

Next time
  - Kernel Regression (Localized)
  - Regression Trees