Lecture 5 – Multivariate Linear Regression

Dan Sheldon

September 19, 2012

Question From Last Time

Can you find matrices A and B such that $BA = I$, but $AB \neq I$?

Question From Last Time

Can you find matrices A and B such that $BA = I$, but $AB \neq I$?

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad B = A^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Question From Last Time

Can you find matrices A and B such that $BA = I$, but $AB \neq I$?

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad B = A^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Check that

$$BA = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I, \qquad AB = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Question From Last Time

Vectors $\mathbf{x}$ and $\mathbf{y}$ are *orthogonal* if their dot product is zero:

$$\mathbf{x}^T\mathbf{y} = \sum_{i=1}^{N} x_i y_i = 0.$$

E.g.,

$$\underbrace{\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}}_{x_1^T} \underbrace{\begin{bmatrix} 1 \\ -0.5 \\ -0.5 \end{bmatrix}}_{x_2} = 0$$

$x_1^T x_2 = 0$

$x_1^T x_1 \neq 1$

$\hat{x}_1 \leftarrow \frac{x_1}{\|x_1\|}$ Check $\|\hat{x}_1\| = 1$

Question From Last Time

$\|x\| = 1$

$\sqrt{x^T x} = 1 \Rightarrow x^T x$

Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k \in \mathbb{R}^n$ be *mutually* orthogonal *unit vectors*

$$\mathbf{x}_i^T \mathbf{x}_j = 0, \quad i \neq j$$
$$\mathbf{x}_i^T \mathbf{x}_i = 1$$

$AA^T \Rightarrow n \times n$

Consider the product

$k \times n$ $\qquad n \times k$

$$C = A^T A = \underbrace{\begin{bmatrix} - & \mathbf{x}_1^T & - \\ & \vdots & \\ - & \mathbf{x}_2^T & - \\ - & \mathbf{x}_k^T & - \end{bmatrix}}_{A^T} \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_k \\ | & | & & | \end{bmatrix}}_{A} = I_k$$

$C_{11} = x_1^T x_1$

$C_{ij} = x_i^T x_j$

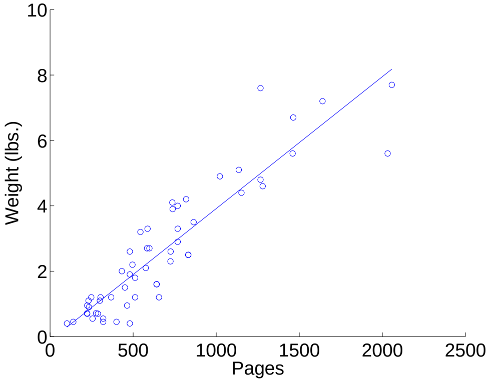
Today's Topics

Multivariate linear regression

- Model
- Cost function
- Normal equations
- Gradient descent
- Features

MATLAB (time permitting)

Book Data



$$y = 0.004036x - .122291$$

Book Data

Can we predict better with multiple features?

Width	Thickness	Height	# Pages	Hardcover	Weight
8	1.8	10	1152	1	4.4
8	0.9	9	584	1	2.7
7	1.8	9.2	738	1	3.9
6.4	1.5	9.5	512	1	1.8

Book Data

Can we predict better with multiple features?

Width	Thickness	Height	# Pages	Hardcover	Weight
8	1.8	10	1152	1	4.4
8	0.9	9	584	1	2.7
7	1.8	9.2	738	1	3.9
6.4	1.5	9.5	512	1	1.8

vectors

scalar

Training data

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$$

Multivariate Linear Regression

- Input: $\mathbf{x} \in \mathbb{R}^d$
- Output: $y \in \mathbb{R}$
- Model (hypothesis class): ?
- Cost function: ?

Model

$h_w(x) = w_0 + w_1 \cdot x$

$x = (x_1, x_2, \ldots, x_d)$

$$h_{\mathbf{w}}(\mathbf{x}) =$$

Model

$\cancel{x_1^2}$ $\cancel{x_1 x_4}$

$$h_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_d x_d$$

Model

w = weight vector $\in \mathbb{R}^{d+1}$

$$h_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_d x_d$$

$$h_{\mathbf{w}}(\mathbf{x}) = \begin{bmatrix} w_0 & w_1 & \ldots & w_d \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$$

Model

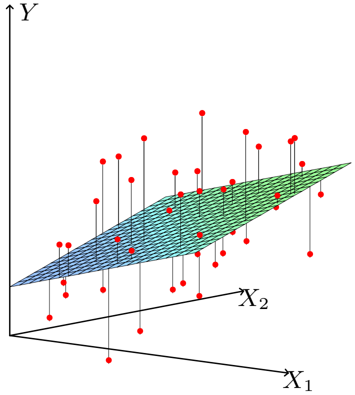
$$h_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_d x_d$$

$$h_{\mathbf{w}}(\mathbf{x}) = \begin{bmatrix} w_0 & w_1 & \ldots & w_d \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$$

$$h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T \mathbf{x} = \mathbf{x}^T \mathbf{w}$$

(Augment feature vector with 1)

Illustration



The Problem

Find $\mathbf{w}$ such that

$$y_i \approx h_{\mathbf{w}}(\mathbf{x}_i), \qquad i = 1, \ldots, N$$

The Problem

$x_{i,d} = d^{th}$ feature of i^{th} example

$x_i^T w$

Find $\mathbf{w}$ such that

$$y_i \approx h_{\mathbf{w}}(\mathbf{x}_i), \qquad i = 1, \ldots, N$$

x_i

$$\begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_N \end{bmatrix} \approx \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \ldots & x_{1,d} \\ 1 & x_{2,1} & x_{2,2} & \ldots & x_{2,d} \\ \ldots & & & & \\ 1 & x_{N,1} & x_{N,2} & \ldots & x_{N,d} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \ldots \\ w_d \end{bmatrix}$$

The Problem

Find $\mathbf{w}$ such that

$$y_i \approx h_{\mathbf{w}}(\mathbf{x}_i), \qquad i = 1, \ldots, N$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_N \end{bmatrix} \approx \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \ldots & x_{1,d} \\ 1 & x_{2,1} & x_{2,2} & \ldots & x_{2,d} \\ \ldots & & & & \\ 1 & x_{N,1} & x_{N,2} & \ldots & x_{N,d} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \ldots \\ w_d \end{bmatrix}$$

$$\mathbf{y} \approx X\mathbf{w}$$

$y_1 \approx x_1^T w$

$r_1 = y_1 - x_1^T w$

$r = Xw - y$

Inputs

Data matrix, label vector

$$X = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,d} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,d} \\ & & \vdots & & \\ 1 & x_{N,1} & x_{N,2} & \dots & x_{N,d} \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}.$$

Width	Thickness	Height	# Pages	Hardcover	Weight
8	1.8	10	1152	1	4.4
8	0.9	9	584	1	2.7
7	1.8	9.2	738	1	3.9
6.4	1.5	9.5	512	1	1.8

X

y

Cost Function

$$J(\mathbf{w}) =$$

Cost Function

$w^T x_i$

$$J(\mathbf{w}) = \sum_{i=1}^{N} (h_{\mathbf{w}}(\mathbf{x}_i) - y_i)^2$$

Exercise: write this succinctly in matrix-vector notation

$= x_i^T w - y_i$

$r_i = h_w(x_i) - y_i$

$r = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix}$

$J(w) = \sum_{i=1}^{N} r_i^2$

$= r^T r = (Xw - y)^T (Xw - y)$

$r = Xw - y$

Cost Function

Answer:

$$J(\mathbf{w}) = (X\mathbf{w} - \mathbf{y})^T (X\mathbf{w} - \mathbf{y})$$

Solution 1: Normal Equations

Normal equations

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

Heuristic derivation:

$$\boxed{\cancel{\begin{aligned} Xw &\approx y \\ X^{-1}Xw &\approx X^{-1}y \\ w &\approx X^{-1}y \end{aligned}}}$$

$$(X^TX)w \approx X^Ty$$

$$(X^TX)^{-1}(X^TX)w \approx (X^TX)^{-1}X^Ty$$

$$w \approx (X^TX)^{-1}X^Ty$$

Proper Approach

- Set all partial derivatives to zero

$$0 = \frac{\partial}{\partial w_j} J(\mathbf{w}) \qquad j = 1, \dots, d$$

- Solve a system of $d + 1$ linear equations for $w_0, \dots, w_d$
- Tedious, but leads to normal equations

Aside: Matrix Calculus

$$\frac{d}{dx} \qquad \frac{\partial}{\partial x}$$

variable

Succinct (and cool!) way to solve for normal equations:

$$0 = \frac{d}{d\mathbf{w}}(X\mathbf{w} - \mathbf{y})^T(X\mathbf{w} - \mathbf{y})$$

Aside: Matrix Calculus

Succinct (and cool!) way to solve for normal equations:

$$
\begin{aligned}
0 &= \frac{d}{d\mathbf{w}}(X\mathbf{w}-\mathbf{y})^T(X\mathbf{w}-\mathbf{y}) \\
0 &= 2(X\mathbf{w}-\mathbf{y})^T X
\end{aligned}
$$

Aside: Matrix Calculus

Succinct (and cool!) way to solve for normal equations:

$$
\begin{aligned}
0 &= \frac{d}{d\mathbf{w}}(X\mathbf{w}-\mathbf{y})^T(X\mathbf{w}-\mathbf{y}) \\
0 &= 2(X\mathbf{w}-\mathbf{y})^T X \\
0 &= X^T(X\mathbf{w}-\mathbf{y})
\end{aligned}
$$

Aside: Matrix Calculus

Succinct (and cool!) way to solve for normal equations:

$$
\begin{aligned}
0 &= \frac{d}{d\mathbf{w}}(X\mathbf{w}-\mathbf{y})^T(X\mathbf{w}-\mathbf{y}) \\
0 &= 2(X\mathbf{w}-\mathbf{y})^T X \\
0 &= X^T(X\mathbf{w}-\mathbf{y}) \\
X^T X\mathbf{w} &= X^T\mathbf{y}
\end{aligned}
$$

Aside: Matrix Calculus

Succinct (and cool!) way to solve for normal equations:

$$
\begin{aligned}
0 &= \frac{d}{d\mathbf{w}}(X\mathbf{w}-\mathbf{y})^T(X\mathbf{w}-\mathbf{y}) \\
0 &= 2(X\mathbf{w}-\mathbf{y})^T X \\
0 &= X^T(X\mathbf{w}-\mathbf{y}) \\
X^T X\mathbf{w} &= X^T\mathbf{y} \\
\mathbf{w} &= (X^T X)^{-1}X^T\mathbf{y}
\end{aligned}
$$

Solution 2: Gradient Descent

1. Initialize $w_0, w_1, \dots, w_d$ arbitrarily
2. Repeat until convergence

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\mathbf{w}), \qquad j = 0, \dots, d.$$

Solution 2: Gradient Descent

1. Initialize $w_0, w_1, \ldots, w_d$ arbitrarily
2. Repeat until convergence

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\mathbf{w}), \qquad j = 0, \ldots, d.$$

Partial derivatives:

j^{th} feature of i^{th} training example

$$\frac{\partial}{\partial w_j} J(\mathbf{w}) = 2 \sum_{i=1}^{N} (h_{\mathbf{w}}(\mathbf{x}_i) - y_i) x_{i,j}$$

W_0

$X_{i,0} = 1$

Feature Normalization

- Features may have very different numeric ranges

Width	Thickness	Height	# Pages	Hardcover	Weight
8	1.8	10	1152	1	4.4
8	0.9	9	584	1	2.7
7	1.8	9.2	738	1	3.9
6.4	1.5	9.5	512	1	1.8

- Advice: normalize your features!
  - Subtract mean (center)
  - Divide by standard deviation (scale)

Feature Normalization

For each feature j, compute

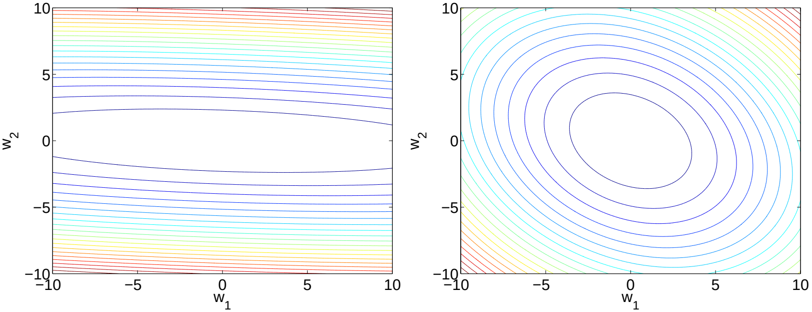
$$\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{i,j}, \qquad \sigma_j^2 = \frac{1}{N}\sum_{i=1}^{N}(x_{i,j} - \mu_j)^2$$

Then, subtract μ_j and divide by σ_j:

$$x_{i,j} \leftarrow (x_{i,j} - \mu_j)/\sigma_j$$

Feature Normalization

Example: cost function contours before and after normalization



Feature Design

It is possible to fit *nonlinear* functions using linear regression:

$$(x_1, x_2, x_3) \mapsto (x_1, x_2, x_3, x_1^2, \log(x_2), x_1 + x_3)$$

Approaches

- Try standard transformations
- Design features you think will work

Polynomial Regression

$w_0 + w_1 x + w_2 x^2 + \ldots$

$$x \mapsto (1, x, x^2, x^3, \ldots)$$