Working with Data & Regression

CS57300 – Data Mining
Purdue University

Instructor: Bruno Ribeiro

Working with Data

Goals:

- Understand How to Work with (Real) Data
- Review Linear Regression
  - Regression as a Naïve Predictor

Data Representation Issues

- How to represent the data?
- E.g.: Are product reviews (stars) really integers between 5 and 1?

Customer Reviews

2,181

3.3 out of 5 stars

5 star	38%
4 star	10%
3 star	11%
2 star	8%
1 star	33%

See all 2,181 customer reviews

Geometric Distribution

- Reviews geometrically distributed?

$$P[X = k|p] = (1-p)^{k-1}p$$

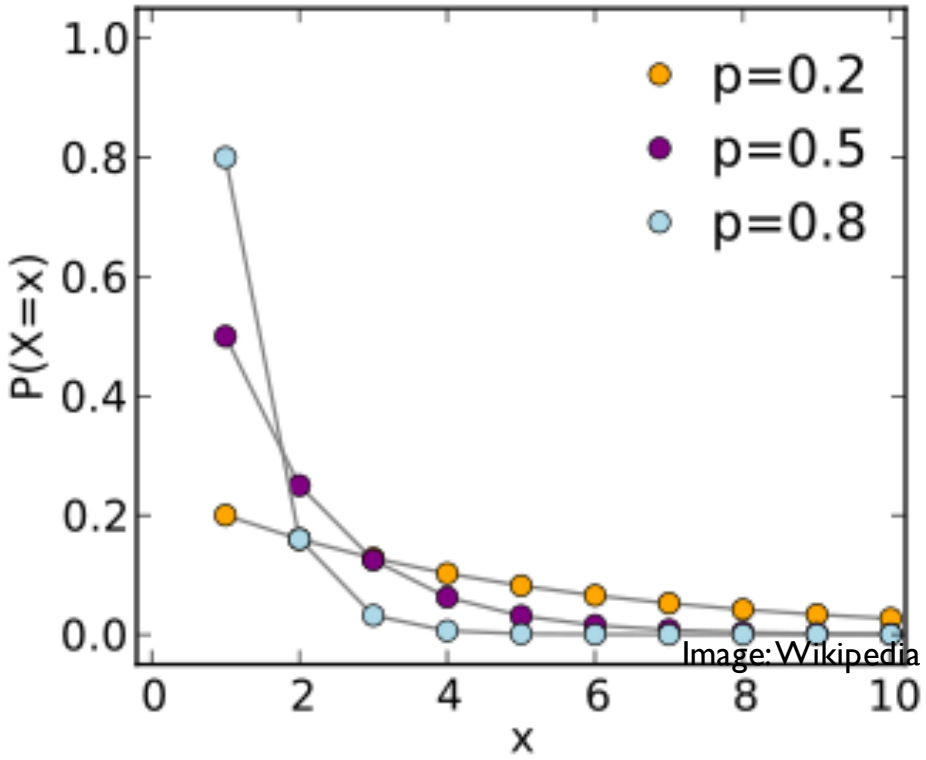
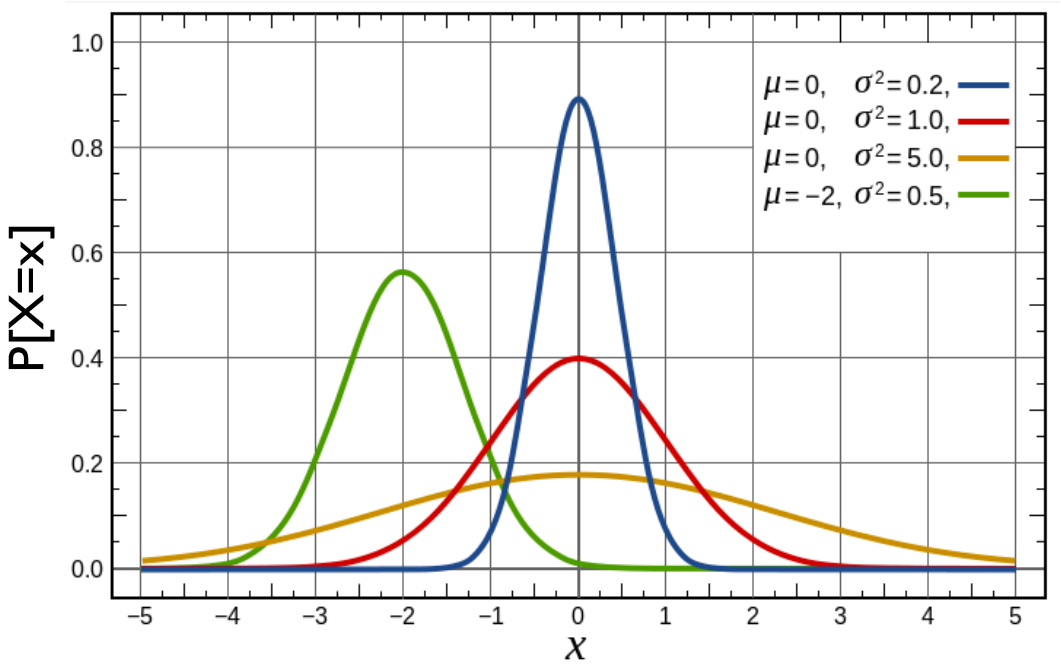


Image: Wikipedia

Normal Distribution

- Reviews normally distributed?

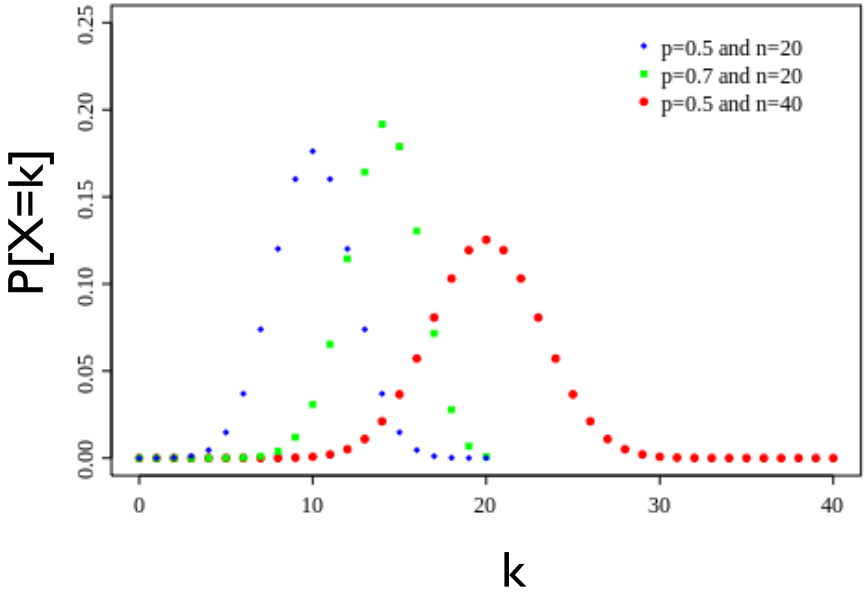
$$P[X = x|\mu, \sigma] = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Binomial distribution

- Reviews binomially distributed?

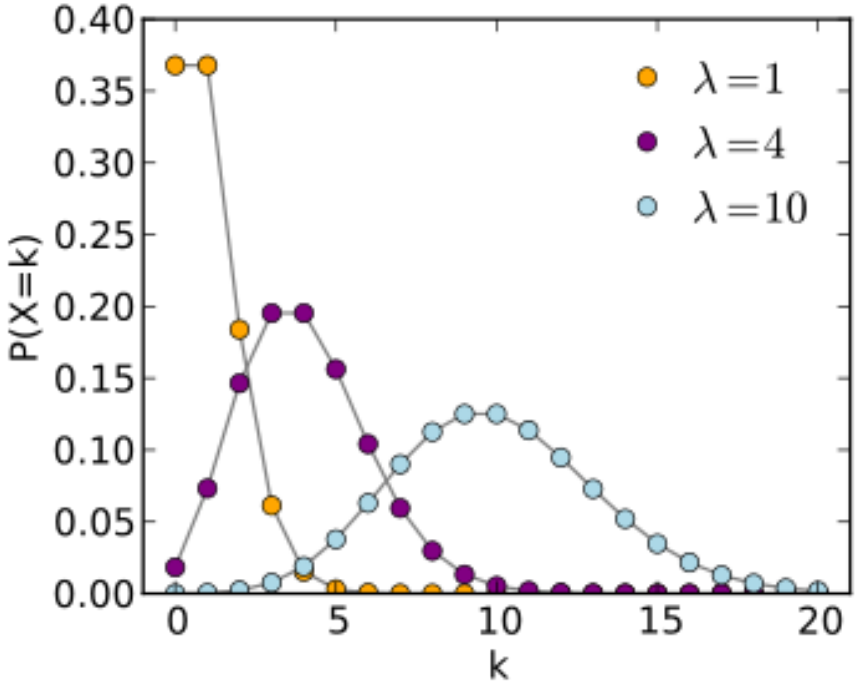
$$P[X = k|p] = \binom{n}{k} p^k (1-p)^{n-k}$$



Poisson distribution

- Reviews Poisson distributed?

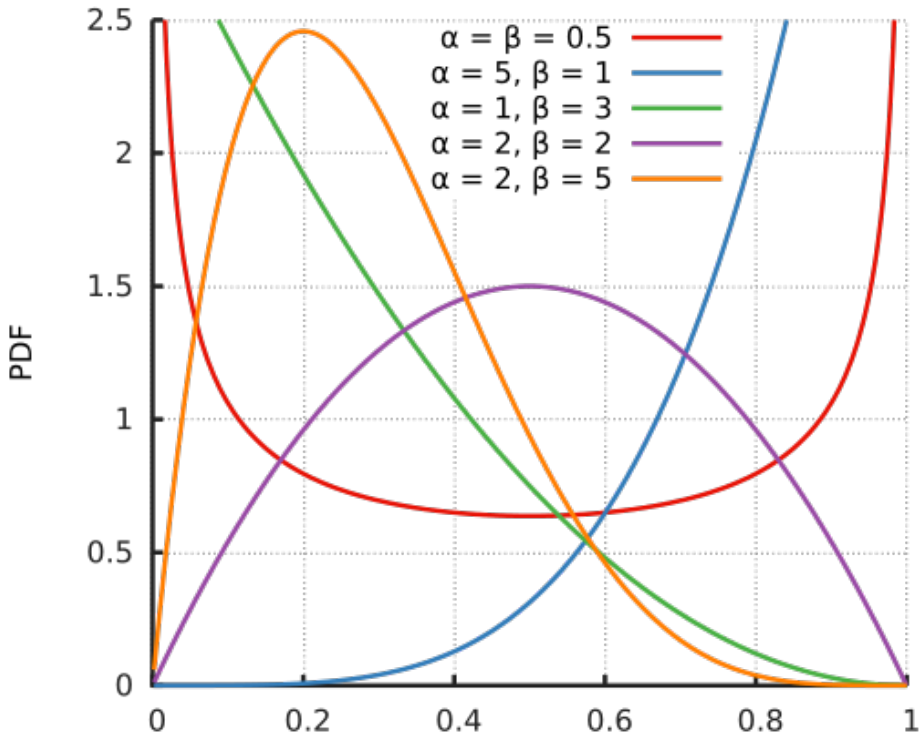
$$P[X = k|\lambda] = \frac{\lambda^k e^{-\lambda}}{k!}$$



Beta distribution

- Reviews beta distributed?

$$P[X = x|\alpha, \beta] = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$



Data representation is key to success

(and a source of headaches if poorly done)

Representing data in Euclidean space

- For datasets with same fixed set of numeric attributes can be thought of as points in a multi-dimensional space, where each dimension represents a distinct attribute

- Many data mining techniques then use similarity/dissimilarity measures to characterize relationships between the instances

Distance measures

- Many data mining techniques utilize similarity/dissimilarity measures to characterize relationships between instances
  - Nearest-neighbor classification
  - Cluster analysis

- **Proximity**: general term to indicate similarity and dissimilarity

- **Distance**: dissimilarity only

Metric properties

- A **metric** *d(x,y)* is a dissimilarity measure that satisfies the following properties:
  - $d(x,y) \geq 0$ *for all* x,y *and* $d(x,y)=0$ *iff* $x=y$ **Positive**
  - $d(x,y) = d(y,x)$ *for all* x,y **Symmetric**
  - $d(x,y) \leq d(x,k)+d(k,y)$ *for all* x,y,k **Triangle inequality**

Distance metrics

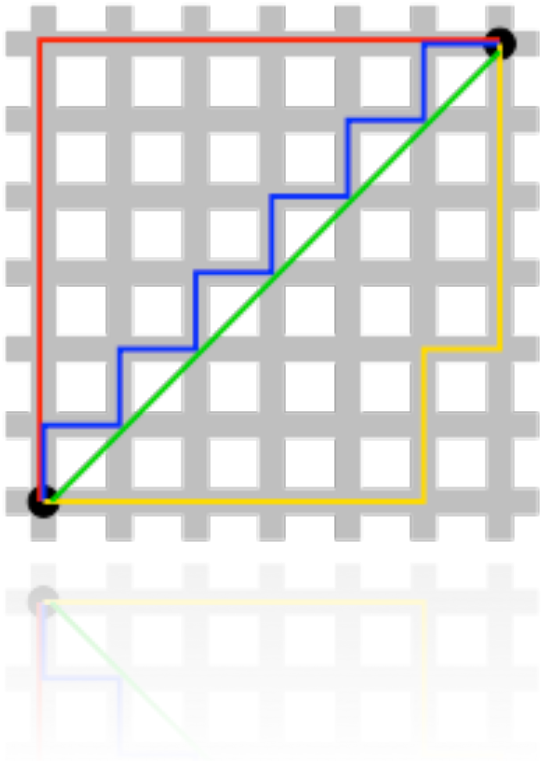
- Manhattan distance (L1)

$$d_M(x,y) = \sum_{i=1}^{k} |x_i - y_i|$$

- Euclidean distance (L2)

$$d_E(x,y) = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

  - Most common metric
  - Assumes variables are commensurate



- **Weighted** Euclidean distance

$$d_E(x,y) = \sqrt{\sum_{i=1}^{k} w_i (x_i - y_i)^2}$$

  - Can weight variables by relative importance

Standardization (dealing with features of different scales)

- Normalization
  - Removes effect of scale
  - Divide each variable by its standard deviation
  - Weights all variables equally

$$x'_i = \frac{x_i - \bar{x}_i}{\hat{\sigma}_i}$$

subtract mean

divide by stdev

$$d_E(x, y) = \sqrt{\sum_{i=1}^{k} (x'_i - y'_i)^2}$$

Features:

Gender: M=0, F=1

Income: 10k – 10,000k

Data:

Employee1 = (0, 30k)

Employee2 = (1,60k)

Without normalization income dominates Euclidean distance

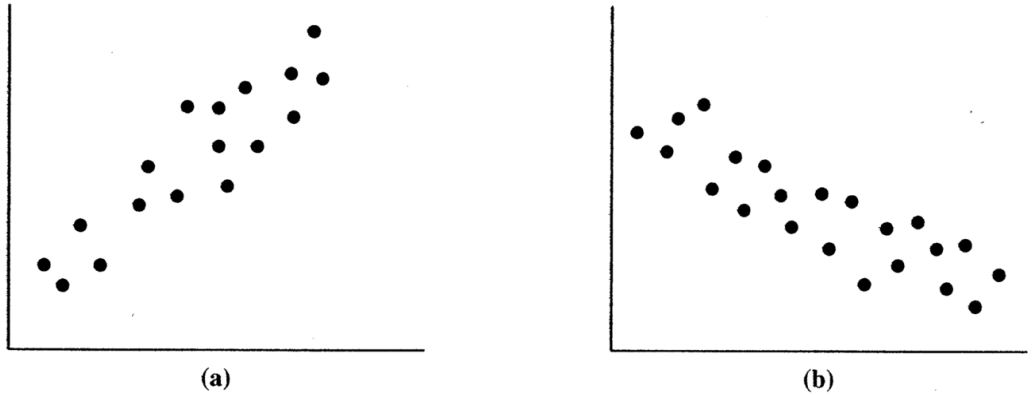
Example of distance measures between random variables: Covariance and correlation

Covariance

- Measures how variables X_j and X_k vary together

$$COV(X_j, X_k) = \frac{1}{n}\sum_{i=1}^{n}\left(x_{ij} - \bar{X}_j\right)\left(x_{ik} - \bar{X}_k\right)$$

  - Positive if large values of X_j are associated with large values of X_k
  - Negative if large values of X_j are associated with small values of X_k



Measures **linear** relationship

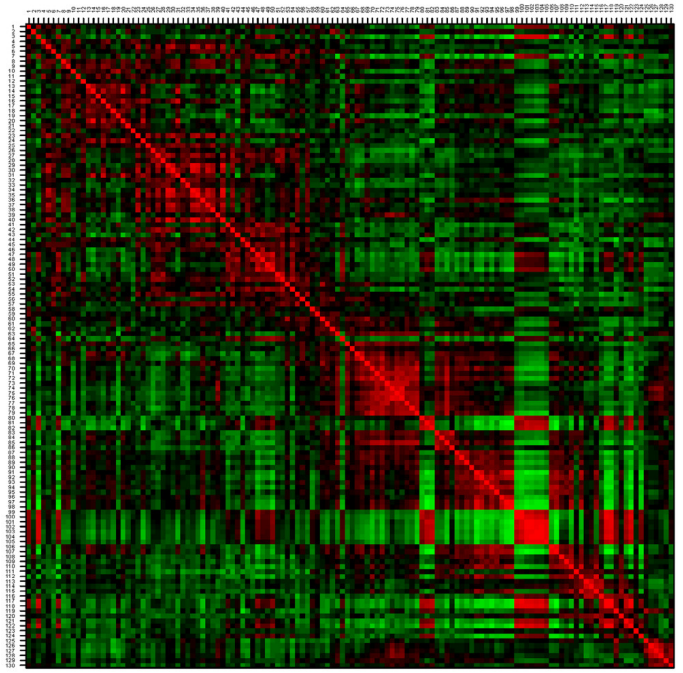
- Covariance matrix (Σ)
  - Symmetric matrix of covariances for p variables

Correlation coefficient

- Covariance depends on ranges of X_j and X_k
- Correlation standardizes covariance by dividing through standard deviations

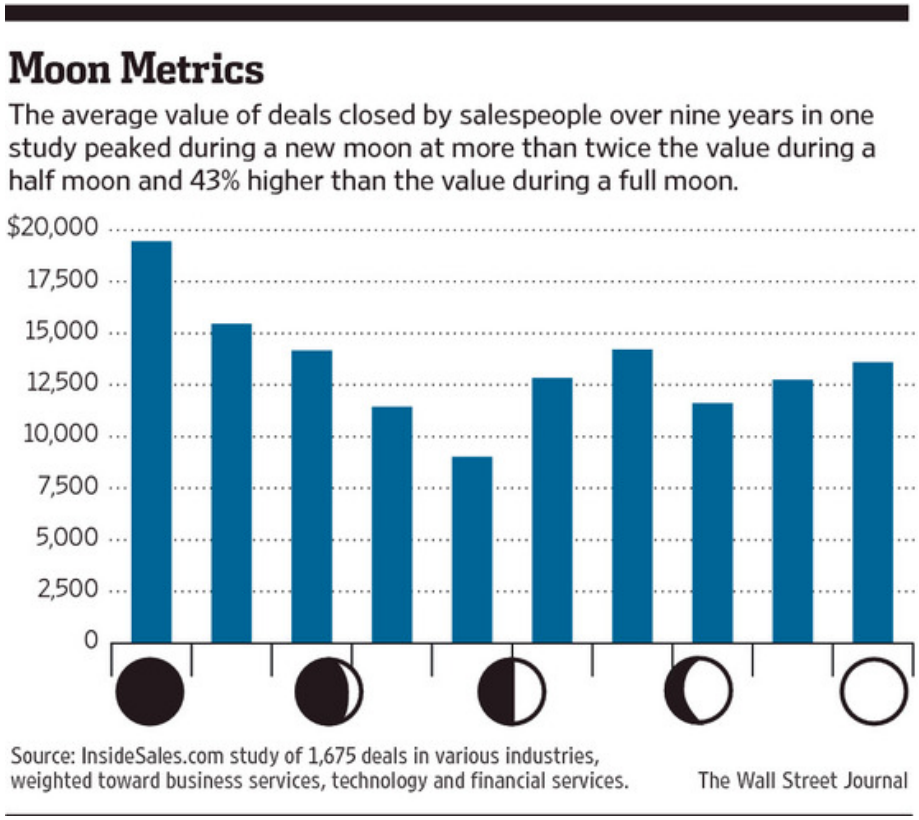
$$\rho(X_j, X_k) = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(x_{ij} - \bar{X}_j\right)\left(x_{ik} - \bar{X}_k\right)}{\sigma_{X_j}\sigma_{X_k}}$$

- Correlation matrix
  - Symmetric matrix of correlations for p variables
  - What values are on the diagonal?



Issues with multidimensional distance metrics?

- Dimensions (features) may be correlated
- E.g.:
  - Gender and income are correlated
  - Some correlations in your data are just plain weird



* not a scientific study

Correlation inflates distance

- Normalization helps little
  if many features are correlated

$$x'_i = \frac{x_i - \bar{x}_i}{\hat{\sigma}_i}$$

subtract mean

divide by stdev

- Solution?

Mahalanobis distance

$$d_{MH}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y})}$$

pxp covariance matrix

- Automatically accounts for scaling
- Corrects for correlation between attributes
- Tradeoff:
  - Covariance matrix can be hard to estimate accurately
  - Memory and time complexity is quadratic rather than linear

Distance measures for binary data

- d(x,y) when items x and y are p-dimensional binary vectors
- Let n_{11} be the number of attributes where both items have value 1, etc.

$$n_{11} = \sum_{i}^{p} \mathbb{I}(x_i + y_i = 2)$$

	y=1	y=0
x=1	n_{11}	n_{11}
x=0	n_{10}	n_{00}

- Matching coefficient
  - Hamming distance normalized by number of bits

$$d_{MC}(x,y) = \frac{n_{11} + n_{00}}{n_{11} + n_{00} + n_{10} + n_{01}}$$

- Jaccard coefficient
  - If we don't care about matches on zeros

$$d_{JC}(x,y) = \frac{n_{11}}{n_{11} + n_{00} + n_{10} + n_{01}}$$

More data issues:

Sometimes data is missing

Missing values

- Reasons for missing values
  - Information is not collected (e.g., people decline to give their age)
  - Attributes may not be applicable to all cases (e.g., annual income is not applicable to children)
- Ways to handle missing values
  - Eliminate entities with missing values
  - Estimate attributes with missing values
  - Ignore the missing values during analysis
  - Replace with all possible values (weighted by their probabilities)
  - Impute missing values

Duplicate Data

- Data set may include data entities that are duplicates, or almost duplicates of one another
  - Major issue when merging data from heterogeneous sources
  - Example: same person with multiple email addresses

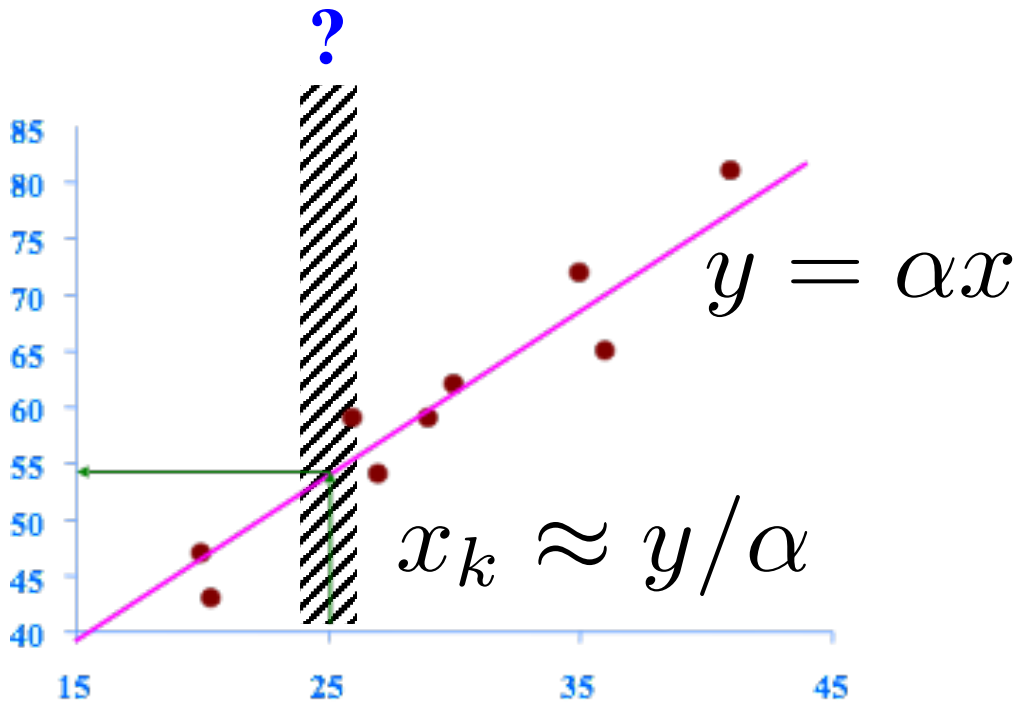
- Data cleaning
  - Finding and dealing with duplicate entities
  - Finding and correcting measurement error
  - Dealing with missing values

Naïve Prediction: Linear Regression

Linear Regression (use A)

- Interpolation
  (something is missing)

- $(x_1, \ldots, x_t)$
- $(y_1, \ldots, y_t)$



Faloutsos 2014

Auto-regression: Predicting Next Value After t Steps
Linear Regression (use B)

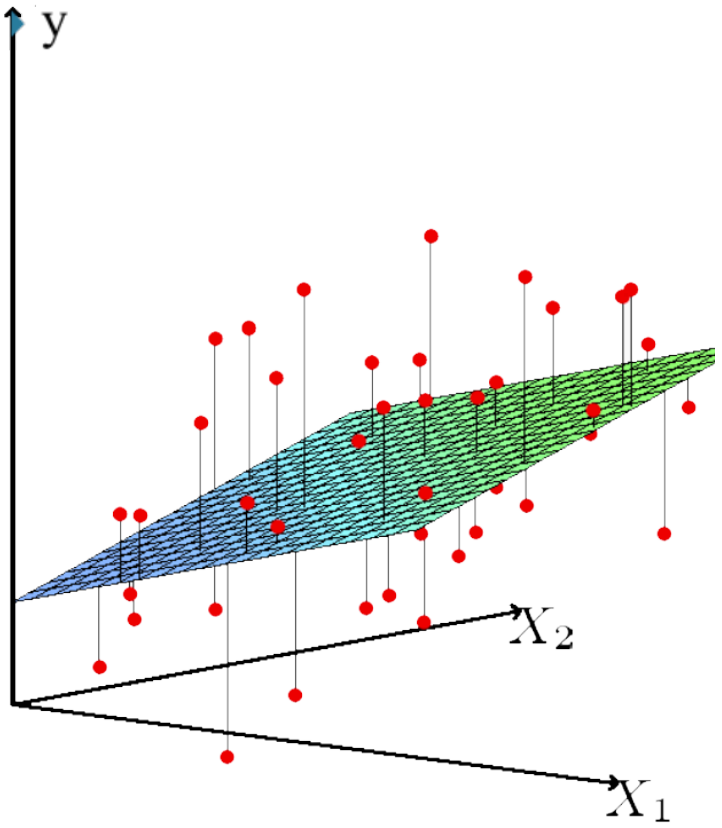
Google Finance

$$x_{t+1} = \sum_{i=1}^{t} a_i x_i + \epsilon_{\text{noise}}$$

Similar problem to linear regression:
express unknowns as a linear function of knowns

Predictions from High-Dimensional Historical Data

- $X_{[t \times w]} \cdot a_{[w \times 1]} = y_{[t \times 1]}$



- Over-constrained problem
  - **a** is the vector of the regression coefficients
  - **X** has the *t* values of the *w* indep. variables
  - **y** has the *t* values of the dependent variable

Ack: C. Faloutosos

Looking Into Multiplication

- $X_{[t \times w]} \cdot a_{[w \times 1]} = y_{[t \times 1]}$

may want to add social media variables

time

$$\begin{bmatrix} X_{11}, X_{12}, \ldots, X_{1w} \\ X_{21}, X_{22}, \ldots, X_{2w} \\ \vdots \\ \vdots \\ \vdots \\ X_{t1}, X_{t2}, \ldots, X_{tw} \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_w \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_t \end{bmatrix}$$

Predicting corn prices over time…

How to Estimate a?

- $\mathbf{a} = (\mathbf{X}^{\mathrm{T}} \cdot \mathbf{X})^{-1} \cdot (\mathbf{X}^{\mathrm{T}} \cdot \mathbf{y})$

$\mathbf{X}^{+} = (\mathbf{X}^{\mathrm{T}} \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^{\mathrm{T}}$ is the Moore–Penrose pseudoinverse

Or: $\mathbf{a} = \mathbf{X}^{+}\, \mathrm{y}$

$\mathbf{a}$ is the vector that minimizes the Root Mean Squared Error (RMSE) of $(\mathbf{y} - \mathbf{X} \cdot \mathbf{a}^{\mathrm{T}})$

Details: Least Squares Optimization

- Least squares cost function:

$$C = \frac{1}{2}\sum_{i=1}^{t}(\mathbf{y}_i - \mathbf{x}_i^T\mathbf{a})^2 = \frac{1}{2}(y - \mathbf{X}a)^T(y - \mathbf{X}a)$$

- Find **a** that minimizes cost C

$$\frac{\partial C}{\partial \mathbf{a}} = \frac{1}{2}\frac{\partial}{\partial \mathbf{a}}(\mathbf{y} - \mathbf{X}\mathbf{a})^T(\mathbf{y} - \mathbf{X}\mathbf{a})$$
$$= -(\mathbf{y} - \mathbf{X}\mathbf{a})^T\mathbf{X}$$

$$\underbrace{\begin{bmatrix} X_{11}, X_{12}, \ldots, X_{1w} \\ X_{21}, X_{22}, \ldots, X_{2w} \\ \vdots \\ \vdots \\ \vdots \\ X_{t1}, X_{t2}, \ldots, X_{tw} \end{bmatrix}}_{\mathbf{X}} \times \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_w \end{bmatrix}}_{\mathbf{a}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_t \end{bmatrix}}_{\mathbf{y}}$$

- Optimal value at:

$$\frac{\partial C}{\partial \mathbf{a}} = 0 \implies \mathbf{X}^T\mathbf{y} = \mathbf{X}^T\mathbf{X}\mathbf{a} \implies \mathbf{a} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

How to Estimate a?

- $\mathbf{a} = (\mathbf{X}^{\mathrm{T}} \cdot \mathbf{X})^{-1} \cdot (\mathbf{X}^{\mathrm{T}} \cdot \mathbf{y})$

$\mathbf{X}^{+} = (\mathbf{X}^{\mathrm{T}} \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^{\mathrm{T}}$ is the Moore–Penrose pseudoinverse

Or: $\mathbf{a} = \mathbf{X}^{+}\, \mathrm{y}$

$\mathbf{a}$ is the vector that minimizes the Root Mean Squared Error (RMSE) of $(\mathrm{y} - \mathbf{X} \cdot \mathbf{a}^{\mathrm{T}})$

Problems:

Matrix $\mathbf{X}$ grows over time & needs matrix inversion

- $\mathrm{O}(\mathrm{t} \cdot \mathrm{w}^2)$ computation
- $\mathrm{O}(\mathrm{t} \cdot \mathrm{w})$ storage

Recursive Least Squares

At time t we know $\mathbf{X}_t = (x_1, \ldots, x_t), \quad \mathbf{y}_t = (y_1, \ldots, y_{t-w})$

Least squares is solving

$$\underset{\mathbf{a}^\star}{\operatorname{argmax}} \|\mathbf{a}^T \mathbf{X}_t - \mathbf{y}_t\|^2$$

which gives

$$\mathbf{a}^\star = \mathbf{X}^+ \mathbf{y}$$

where $\mathbf{X}^+ = (\mathbf{X}^\mathrm{T} \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^\mathrm{T}$

Let

$$\mathbf{\Phi}_t = \mathbf{X}_t^T \mathbf{X}_t \qquad \theta_t = \mathbf{X}_t^T y_t$$

Then $\mathbf{\Phi}_{t+1}^{-1}$ is

$$\mathbf{\Phi}_{t+1}^{-1} = (\mathbf{\Phi}_t + \mathbf{x}_{t+1}\mathbf{x}_{t+1}^T)^{-1} = \mathbf{\Phi}_t^{-1} - \frac{\mathbf{\Phi}_t^{-1}\mathbf{x}_{t+1}\mathbf{x}_{t+1}^T\mathbf{\Phi}_t^{-1}}{1 + \mathbf{x}_{t+1}^T\mathbf{\Phi}_t^{-1}\mathbf{x}_{t+1}}$$

Matrix Inversion Formula

If A and B are $m \times m$ positive definite matrices, D is a $n \times n$ matrix, and C is a $m \times n$ matrix such that

$$A = B^{-1}CD^{-1}C^T\,,$$

then

$$A^{-1} = B - BC(D + C^T BC)^{-1}C^T B.$$

Recursive Least Squares Algorithm

$$\mathbf{\Phi}_{t+1}^{-1} = \mathbf{\Phi}_{t}^{-1} - \frac{\mathbf{\Phi}_{t}^{-1}\mathbf{x}_{t+1}^{T}\mathbf{x}_{t+1}^{T}\mathbf{\Phi}_{t}^{-1}}{1 + \mathbf{x}_{t+1}^{T}\mathbf{\Phi}_{t}^{-1}\mathbf{x}(t+1)}$$

$$\theta_{t+1} = \theta_t + \mathbf{x}_{t+1}^{T}\mathbf{y}_{t+1}$$

$$\mathbf{a}_{t+1} = \mathbf{\Phi}_{t+1}^{-1}\theta_{t+1}$$

Exponentially Weighted Recursive Least Squares Algorithm

for $\lambda > 1$

$$\mathbf{\Phi}_{t+1}^{-1} = \frac{1}{\lambda}\mathbf{\Phi}_t^{-1} - \frac{1}{\lambda^2}\frac{\mathbf{\Phi}_t^{-1}\mathbf{x}_{t+1}^T\mathbf{x}_{t+1}^T\mathbf{\Phi}_t^{-1}}{1+\mathbf{x}_{t+1}^T\mathbf{\Phi}_t^{-1}\mathbf{x}(t+1)}$$

$$\theta_{t+1} = \lambda\theta_t + \mathbf{x}_{t+1}^T\mathbf{y}_{t+1}$$

$$\mathbf{a}_{t+1} = \mathbf{\Phi}_{t+1}^{-1}\theta_{t+1}$$

Comparison

Original Least Squares

- Needs large matrix (growing in size) $O(t \times w)$
- Costly matrix operation $O(t \times w^2)$

Recursive LS

- Need much smaller, fixed size matrix $O(w \times w)$
- Fast, incremental computation $O(1 \times w^2)$
- no matrix inversion

Other data preprocessing methods

- Sampling
- Dimensionality reduction
- Attribute transformation (e.g., discretization, distance calculations)
- Feature construction and selection
- We discuss these in more detail throughout the course