COLUMBIA UNIVERSITY

Statistical NLP for the Web

Introduction, Text Mining, Linear Methods of Regression

Sameer Maskey

Week 1, September 5, 2012

Outline

- Introduction
- Final Project Details
- NLP-ML Topics
- Text Mining, Scoring chunks of text
- Simple Linear Regression
- Multiple Linear Regression
- Equations to Implementation
- Reading Assignment

Course Information

- Course Website: http://www.cs.columbia.edu/~smaskey/CS6998-0412
- Discussions in courseworks
- Office hours Wed: 2 to 4pm, 457 CS building
- Individual appointments in person or in phone can be set by emailing the instructor : smaskey@cs.columbia.edu

- Instructor: Sameer Maskey, PhD
  - Adj. Assistant Professor, Columbia University
  - Research Scientist, IBM Research
  - NLP/Speech Processing research for the last 12 years

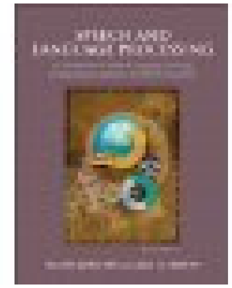
- TA : Morgan Ulinski mulinski@cs.columbia.edu
- Office hours : 2-4pm Tuesday, Speech Lab, CEPSR

- Prerequisites
  - Probability, statistics, linear algebra, programming skill
  - CS Account

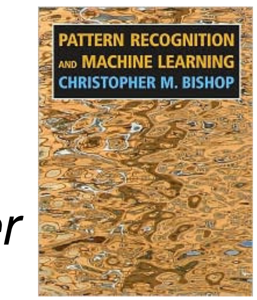
Grading and Academic Integrity

- **3 Homework (15% each)**
  - Homework due dates are available in the class webpage
  - You have 3 ‘no penalty’ late days <u>in total</u> that can be used during the semester
  - Each additional late day (without approval) will be penalized, 20% each day
- No Midterm exam
- **Final project (55%)**
  - It is meant for you to explore and do research NLP/ML topic of your choice
  - Project proposal due soon
- No Final Exam
- Collaboration allowed but presenting someone else’s work (including code) will result in an automatic zero

Textbooks

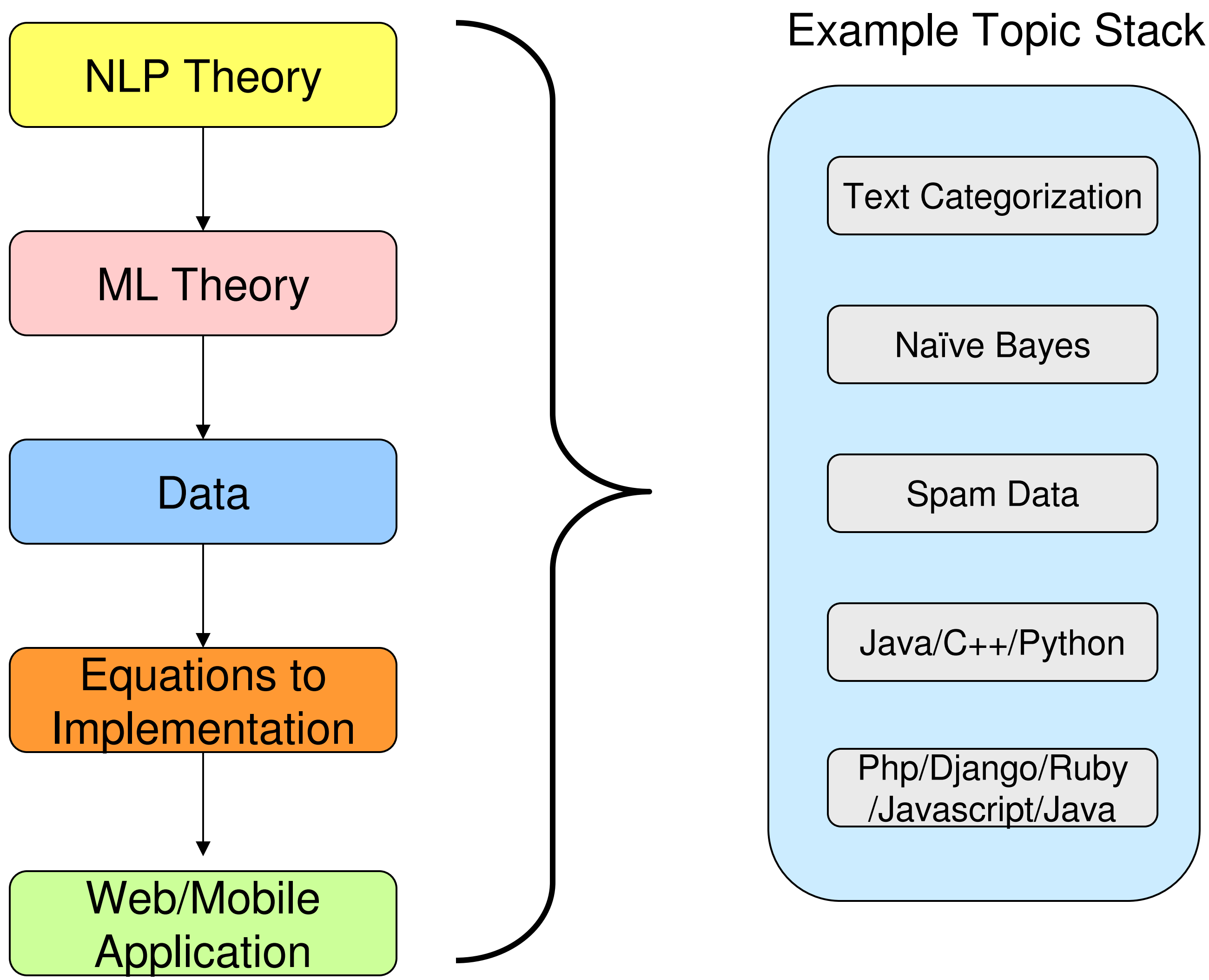


- For NLP topics we will use the following book:
  - *Speech and Language Processing (2nd Edition) by Daniel Jurafsky and James H Martin*



- For statistical methods/ML topics we will partly use
  - *Pattern Recognition and Machine Learning by Christopher Bishop*

How Will We Approach Topics?

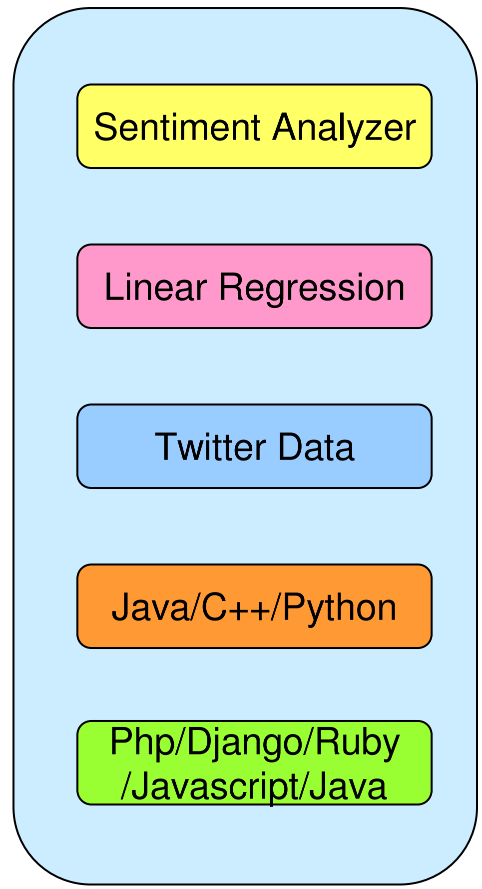


Final Project Details

- 1 to 2 Person Team
- Think about a cool NLP/ML based web/mobile application you have been wanting to build forever and share to the world! (Build it and get course credit for it)

- Project Proposal
- Project 1/2 semester report
- Project 3/4 semester demo
- Project DEMO Day
  - Paper (8 pages)
  - Poster/Slides
  - Demo

Example Final Project

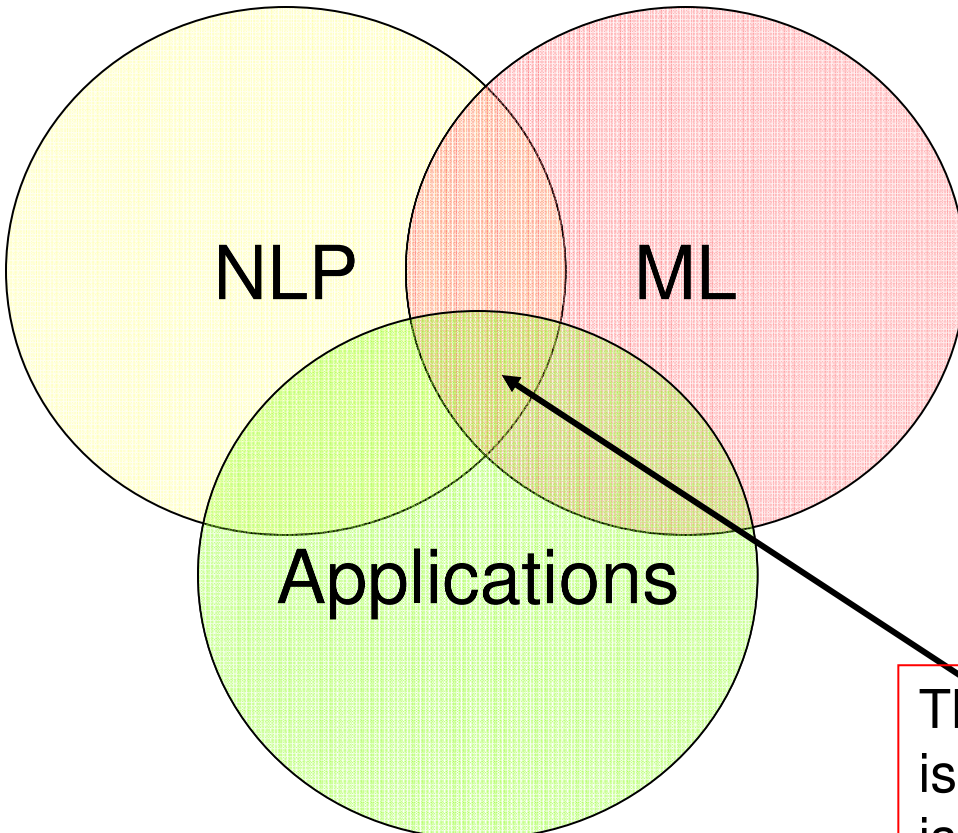


Final Project DEMO/Mini-Conference Day

- December 12, 2012
- Each student/group will present his/her/their work
  - Paper
  - Poster
  - Demo
- Judges
  - Internal Judges
  - External Industry Experts

Computing Environment

- Cs699804.cs.columbia.edu
- Each student will get 15-20 GB of space for experiment
- You can also use CS cluster
- You can also use your laptops/desktops



This intersection is what the class is about

Goal of the Class

- By the end of the semester
  - You will have in-depth knowledge of several NLP and ML topics and explore the relationship between them

  - You should be able to implement many of the NLP/ML methods on your own

  - You will be able to frame many of the NLP problems in statistical framework of your choice

  - You will understand how to analytically read NLP/ML papers and know the kind of questions to ask oneself when doing NLP/ML research

  - You will have built one end-to-end NLP application that hopefully you will be proud of!

Topics in NLP (HLT, ACL) Conference

- Morphology (including word segmentation)
- Part of speech tagging
- Syntax and parsing
- Grammar Engineering
- Word sense disambiguation
- Lexical semantics
- Mathematical Linguistics
- Textual entailment and paraphrasing
- Discourse and pragmatics
- Knowledge acquisition and representation
- Noisy data analysis
- Machine translation
- Multilingual language processing
- Language generation
- Summarization
- Question answering
- Information retrieval
- Information extraction
- Topic classification and information filtering
- Non-topical classification (sentiment/genre analysis)
- Topic clustering
- Text and speech mining
- Text classification
- Evaluation (e.g., intrinsic, extrinsic, user studies)
- Development of language resources
- Rich transcription (automatic annotation)
- …

Topics in ML (ICML, NIPS) Conference

- Reinforcement Learning
- Online Learning
- Ranking
- Graphs and Embeddding
- Gaussian Processes
- Dynamical Systems
- Kernels
- Codebook and Dictionaries
- Clustering Algorithms
- Structured Learning
- Topic Models
- Transfer Learning
- Weak Supervision
- Learning Structures
- Sequential Stochastic Models
- Active Learning
- Support Vector Machines
- Boosting
- Learning Kernels
- Information Theory and Estimation
- Bayesian Analysis
- Regression Methods
- Inference Algorithms
- Analyzing Networks & Learning with Graphs
- …

Many Topics Related
Tasks ←→ Solutions
Combine Relevant Topics

NLP

- Morphology (including word segmentation)
- Part of speech tagging
- Syntax and parsing
- Grammar Engineering
- Word sense disambiguation
- Lexical semantics
- Mathematical Linguistics
- Textual entailment and paraphrasing
- Discourse and pragmatics
- Knowledge acquisition and representation
- Noisy data analysis
- Machine translation
- Multilingual language processing
- Language generation
- Summarization
- Question answering
- Information retrieval
- Information extraction
- Topic classification and information filtering
- Non-topical classification (sentiment/genre analysis)
- Topic clustering
- Text and speech mining
- Text classification
- Evaluation (e.g., intrinsic, extrinsic, user studies)
- Development of language resources
- Rich transcription (automatic annotation)
- …

ML

- Reinforcement Learning
- Online Learning
- Ranking
- Graphs and Embeddding
- Gaussian Processes
- Dynamical Systems
- Kernels
- Codebook and Dictionaries
- Clustering Algorithms
- Structured Learning
- Topic Models
- Transfer Learning
- Weak Supervision
- Learning Structures
- Sequential Stochastic Models
- Active Learning
- Support Vector Machines
- Boosting
- Learning Kernels
- Information Theory and Estimation
- Bayesian Analysis
- Regression Methods
- Inference Algorithms
- Analyzing Networks & Learning with Graphs
- …

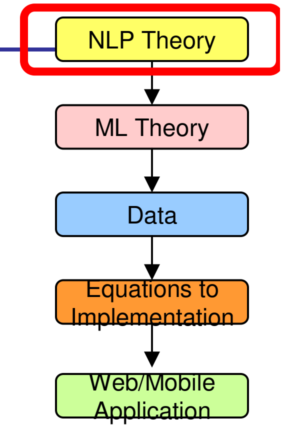
Topics We Will Cover in This Course

NLP -- ML

NLP	ML
▪ Text Mining	Linear Models of Regression
▪ Text Categorization	Linear Methods of Classification Generative Classifier
▪ Information Extraction/Tagging	Hidden Markov Model Maximum Entropy Models
▪ Syntax and Parsing	Viterbi Search, Beam Search
▪ Topic and Document Clustering	K-means, KNN Expectation Maximization
▪ Machine Translation	
▪ Language Modeling	Neural Networks
▪ Evaluation Techniques	
	Deep Belief Networks Belief Propogation

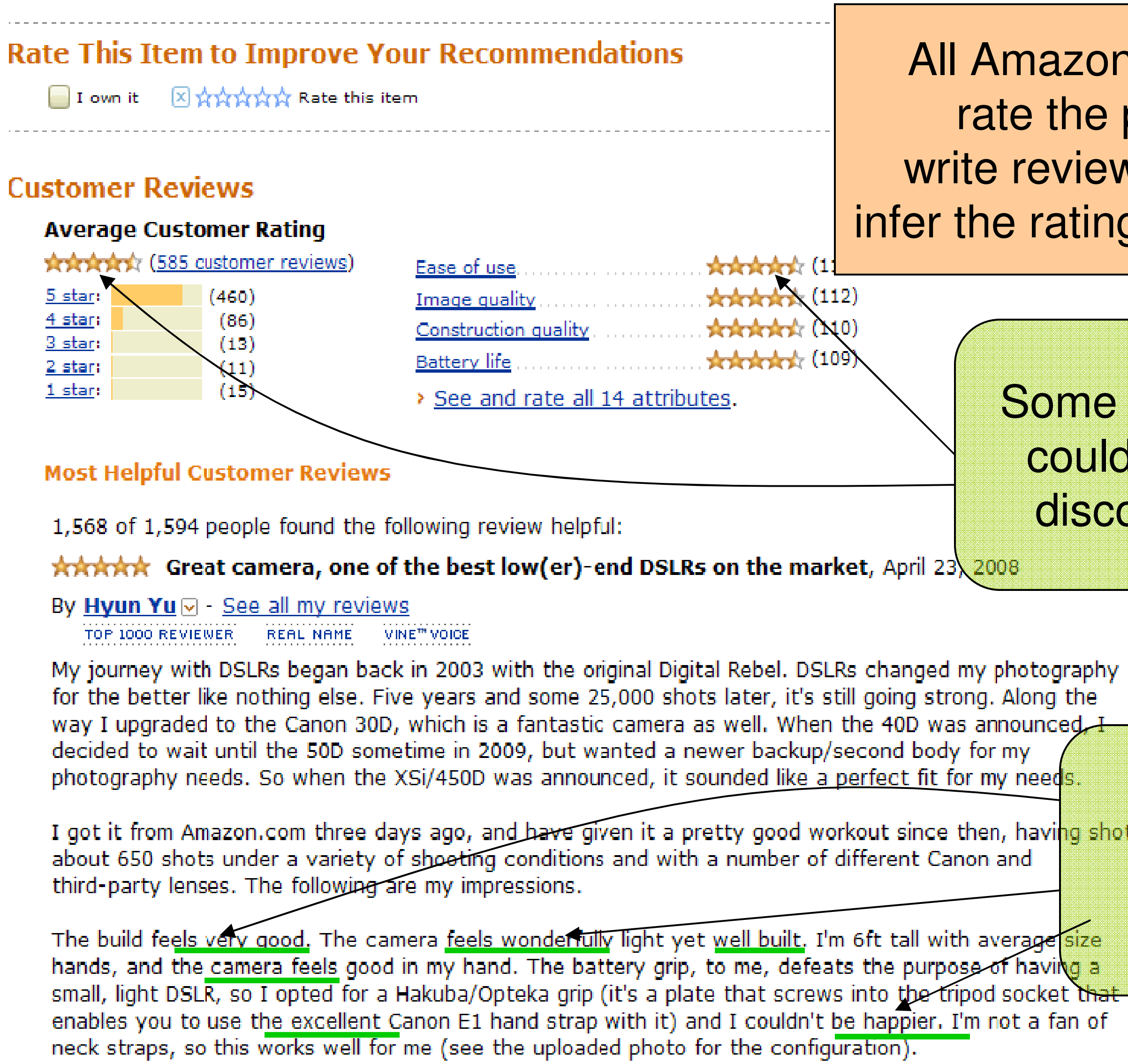
How about Data?

Text Mining



- Data Mining: finding nontrivial patterns in corpora/databases that may be previously unknown and could be useful
- Text Mining:
  - Find interesting patterns/information from unstructured text
  - Discover new knowledge from these patterns/information
- Information Extraction, Summarization, Opinion Analysis, etc can be thought as some form of text mining
- Let us look at an example

Patterns in Unstructured Text


Rate This Item to Improve Your Recommendations

I own it   Rate this item

Customer Reviews

Average Customer Rating

(585 customer reviews)

5 star:		(460)
4 star:		(86)
3 star:		(13)
2 star:		(11)
1 star:		(15)

Ease of use	(1
Image quality	(112)
Construction quality	(110)
Battery life	(109)

› See and rate all 14 attributes.

Most Helpful Customer Reviews

1,568 of 1,594 people found the following review helpful:

Great camera, one of the best low(er)-end DSLRs on the market, April 23, 2008

By **Hyun Yu** - See all my reviews

TOP 1000 REVIEWER   REAL NAME   VINE™ VOICE

My journey with DSLRs began back in 2003 with the original Digital Rebel. DSLRs changed my photography for the better like nothing else. Five years and some 25,000 shots later, it's still going strong. Along the way I upgraded to the Canon 30D, which is a fantastic camera as well. When the 40D was announced, I decided to wait until the 50D sometime in 2009, but wanted a newer backup/second body for my photography needs. So when the XSi/450D was announced, it sounded like a perfect fit for my needs.

I got it from Amazon.com three days ago, and have given it a pretty good workout since then, having shot about 650 shots under a variety of shooting conditions and with a number of different Canon and third-party lenses. The following are my impressions.

The build feels very good. The camera feels wonderfully light yet well built. I'm 6ft tall with average size hands, and the camera feels good in my hand. The battery grip, to me, defeats the purpose of having a small, light DSLR, so I opted for a Hakuba/Opteka grip (it's a plate that screws into the tripod socket that enables you to use the excellent Canon E1 hand strap with it) and I couldn't be happier. I'm not a fan of neck straps, so this works well for me (see the uploaded photo for the configuration).

All Amazon reviewers may not rate the product, may just write reviews, we may have to infer the rating based on text review

Some of these patterns could be exploited to discover knowledge

Patterns may exist in unstructured text

Review of a camera in Amazon

Text to Knowledge

- Text
  - Words, Reviews, News Stories, Sentences, Corpus, Text Databases, Real-time text, Books

Many methods to use for discovering knowledge from text

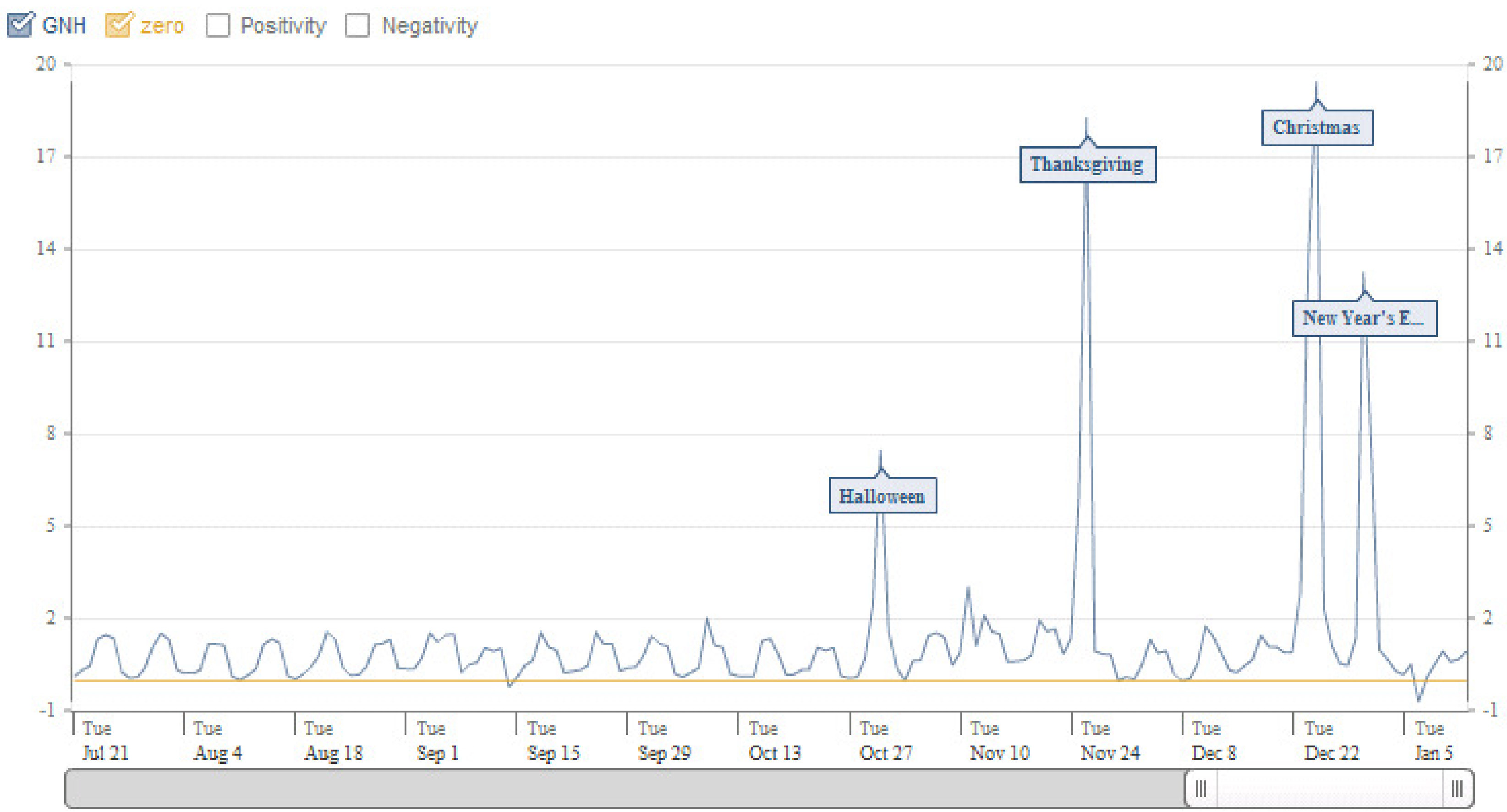
- Knowledge
  - Ratings, Significance, Patterns, Scores, Relations

Unstructured Text → Score
Facebook's "Gross National Happiness Index"

- Facebook users update their status
  - "…is writing a paper"
  - "… has flu ☹"
  - "… is happy, yankees won!"

- Facebook updates are unstructured text
- Scientists collected all updates and analyzed them to predict "Gross National Happiness Index"

Facebook's "Gross National Happiness Index"



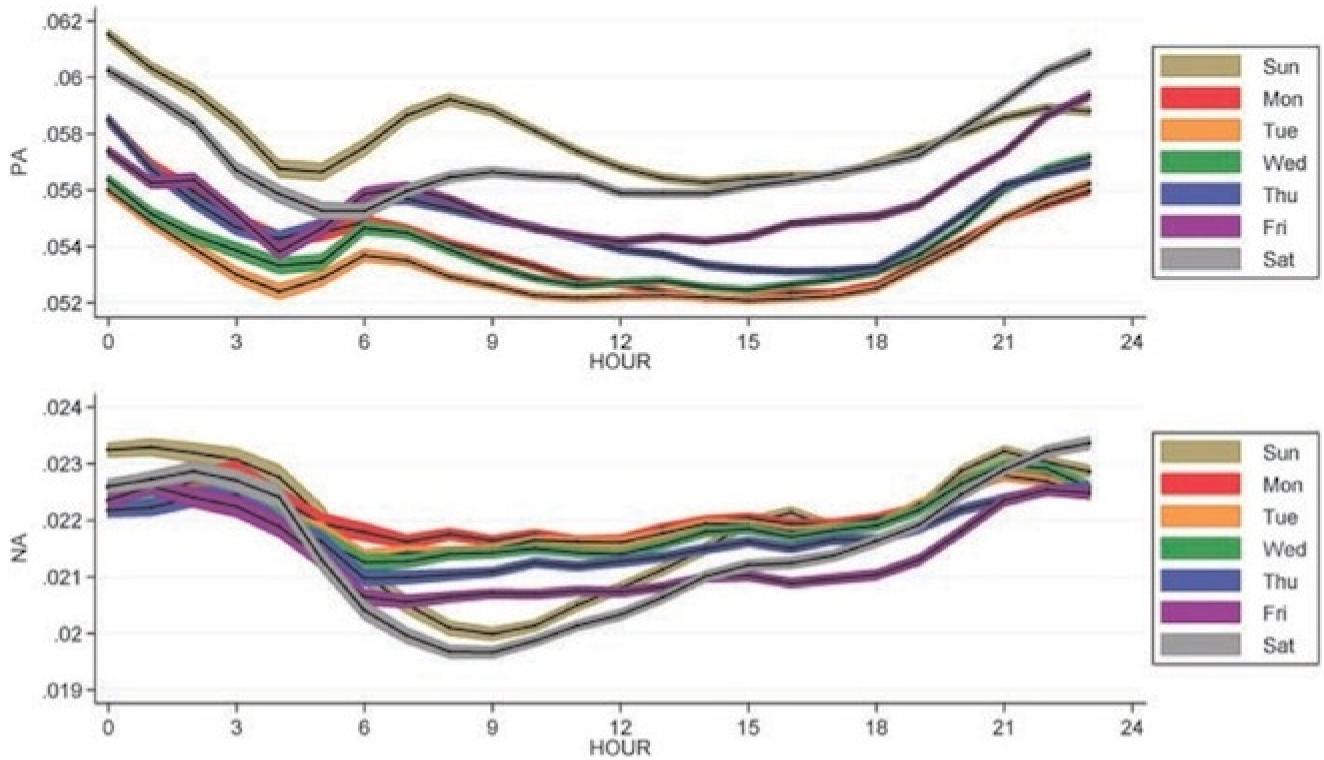
How do you think they extracted this SCORE from a TEXT collection of status updates?

Facebook Blog Explains

- “The result was an index that measures how happy people on Facebook are from day-to-day by looking at the number of positive and negative words they're using when updating their status. When people in their status updates use more positive words - or fewer negative words - then that day as a whole is counted as happier than usual.”

Looks like they are COUNTING!
+ve and –ve words in status updates

Mood Swings During a Day Based on Twitter Data



- Tweets → Score
- 509 Million tweets analyzed
- 2.4 Million Users

"Diurnal and Seasonal Mood Vary with Work, Sleep, and Daylength Across Diverse Cultures." By Scott A. Golder and Michael W. Macy. Science, Vol. 333, September 30, 2011.

Let’s Build Our NLP/ML Model to Predict Happiness ☺

- Simple Happiness Score
  - Our simpler version of happiness index compared to facebook
  - Score ranges from 0 to 10

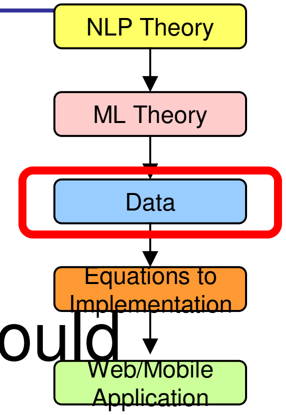
- There are a few things we need to consider
  - We are using status updates words
  - We do not know what words are positive and negative
  - We do not have any training data

Our Prediction Problem

- <u>Training data</u>
  - Assume we have N=100,000 status updates
  - Assume we have a simple list of positive and negative words
  - Let us also assume we asked a human annotator to read each of the 100,000 status update and give a happiness Score (Y_i) between 0 to 10
    - "…is writing a paper" ($Y_1 = 4$)
    - "… has flu ☹" ($Y_2 = 1.8$)
    - .
    - .
    - .
    - "… is happy, game was good!" ($Y_{100,000} = 8.9$)
- <u>Test data</u>
  - **"… likes the weather" ($Y_{100,001}$ = ?)**

Given labeled set of 100K Status updates, how do we build Statistical/ML model that will predict the score for a new status update

Representing Text of Status Updates As a Vector



- What kind of feature can we come up with that would relate well with happiness score

- How about represent status update as
  - Count (+ve words in the sentence) (not the ideal representation, will see better representation letter)
  - For the 100,000th sentence in our previous example:
    - "…is happy, game was good." Count is 2
    - Status Update 100,000th is represented by
      - $(X_{100000} = 2, Y_{100000} = 8.9)$

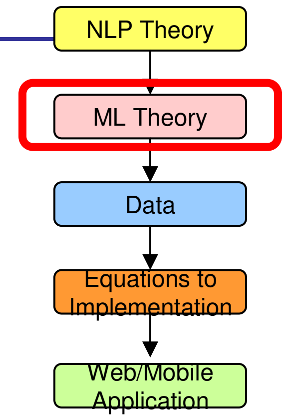
Modeling Technique

- We want to predict happiness score (Y_i) for a new status update
- If we can model our training data with a statistical/ML model, we can do such prediction
  - (1, 4)
  - (0, 1.8)
  - .
  - .
  - .
  - (2, 8.9)

$\left(\boxed{Xi} \,,\, \boxed{Yi} \right)$

- What modeling technique can we use?
  - Linear Regression is one choice

Linear Regression



→We want to find a function that given our x it would map it to y

→One such function :

$$f(x) = \theta_0 + \theta_1 x$$

→Different values of thetas give different functions

→What is the best theta such that we have a function that makes <u>least error</u> on predictions when compared with y

For notation convenience we may use $f(x)$ dropping θ in $f_\theta(x)$ or $f(x; \theta)$ when it is understood

Predicted vs. True

- Our function $f(x)$ approximates y
- Given a true value of y we can compute the error $f(x)$ made against true y
- For any point x_i we can compute such error by $y_i - f(x_i)$; or by squared error $\{y_i - f(x_i)\}^2$
- But we have N points so the total error/Loss L on squared error would be

$$L = \sum_{i=1}^{N} (y_i - f(x_i))^2$$

Sum of Squared Errors

- Plugging in f(x) and averaging the error across all training data points we get the empirical loss

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (\theta_0 + \theta_1 x_i))^2$$

Finding the Minimum

→We can (but not always) find a minimum of a function by setting the derivative or partial derivatives to zero

→Here we can take partials on thetas and set them to zero

$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{\partial}{\partial \theta_0} \frac{1}{2N} \sum_{i=1}^{N} (y_i - (\theta_0 + \theta_1 x_i))^2$$

$$\frac{\partial J(\theta)}{\partial \theta_1} = \frac{\partial}{\partial \theta_1} \frac{1}{2N} \sum_{i=1}^{N} (y_i - (\theta_0 + \theta_1 x_i))^2$$

Solving for Weights

$$\begin{aligned}
\frac{\partial J(\theta)}{\partial \theta_1} &= \frac{\partial}{\partial \theta_1}\frac{1}{2N}\sum_{i=1}^{N}(y_i - (\theta_0 + \theta_1 x_i))^2 \\
&= \frac{1}{2N}\sum_{i=1}^{N}\frac{\partial}{\partial \theta_1}(y_i - \theta_0 - \theta_1 x_i)^2 \\
&= \frac{1}{2N}\sum_{i=1}^{N}2(y_i - \theta_0 - \theta_1 x_i)\frac{\partial}{\partial \theta_1}(y_i - \theta_0 - \theta_1 x_i) \\
&= \frac{1}{N}\sum_{i=1}^{N}(y_i - \theta_0 - \theta_1 x_i)(-x_i) = 0 \\
\frac{\partial J(\theta)}{\partial \theta_0} &= \frac{\partial}{\partial \theta_0}\frac{1}{2N}\sum_{i=1}^{N}(y_i - (\theta_0 + \theta_1 x_i))^2 \\
&= \frac{1}{N}\sum_{i=1}^{N}(y_i - \theta_0 - \theta_1 x_i)(-1) = 0
\end{aligned}$$

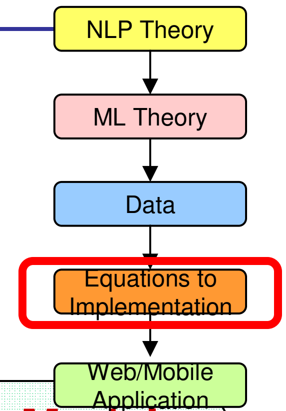
Empirical Loss is Minimized With Given Values for the Parameters

→Solving the previous equations we get following values for the thetas

$$\theta_1 = \frac{\sum_{i=1}^{N} x_i y_i - \frac{1}{N}\sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{\sum_{i=1}^{N} x_i^2 - \frac{1}{N}\sum_{i=1}^{N} x_i \sum_{i=1}^{N} x_i}$$

$$\theta_0 = \frac{1}{N}\sum_{i=1}^{N} y_i - \frac{1}{N}\theta_1 \sum_{i=1}^{N} x_i$$

Equations to Implementation



- Given our training data on status update with happiness score
  - (1, 4)
  - (0, 1.8)
  - .
  - .
  - .
  - (2, 8.9)

[Xi , Yi]

Training Our Regression Model:
Just need to implement for loop that computes numerators and denominators in equations here. And we get optimal thetas

$$\theta_1 = \frac{\sum_{i=1}^{N} x_i y_i - \frac{1}{N}\sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{\sum_{i=1}^{N} x_i^2 - \frac{1}{N}\sum_{i=1}^{N} x_i \sum_{i=1}^{N} x_i}$$

For Prediction/Testing:
Given optimal thetas, plug in the x value in our equation to get y

$$\theta_0 = \frac{1}{N}\sum_{i=1}^{N} y_i - \frac{1}{N}\theta_1 \sum_{i=1}^{N} x_i$$

Simple Happiness Scoring Model too Simple?

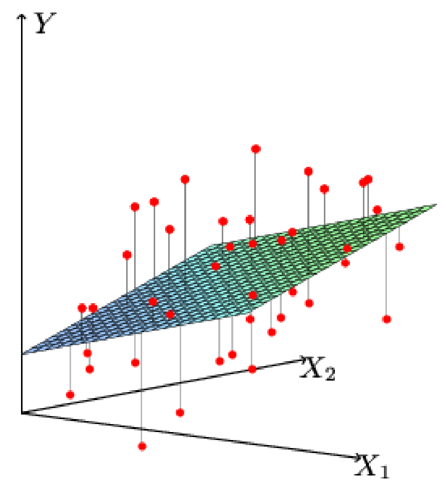
- So far we have a regression model that was trained on a training data of facebook status updates (text) and labeled happiness score
- Status updates words were mapped to one feature
  - Feature counted number of +ve words
- Maybe too simple?
  - How can we improve the model?
  - Can we add more features?
    - How about count of –ve words as well

Let Us Add One More Feature

- Adding one more feature Z_i representing count of –ve words, now training data will look like the following
  - (1, 3, 4)
  - (0, 6,1.8)
  - .
  - .
  - .
  - (2, 0, 8.9)

[Xi , Zi , Yi]

- What would our linear regression function would look like



$$f(x, z) = \theta_0 + \theta_1 x + \theta_2 z$$

Figure 3.1: *Linear least squares fitting with $X \in \mathbb{R}^2$. We seek the linear function of X that minimizes the sum of squared residuals from Y.* [3]

Estimation of y i.e. f(x,z) is now a plane instead of a line

Regression Function in Matrix Form

- Remember our regression function in 2D looked like

$$f(x) = \theta_0 + \theta_1 x$$

- Representing in Matrix form we get,

$$\text{f(x)} = \left[\begin{array}{cc} 1 & x \end{array}\right] . \left[\begin{array}{c} \theta_0 \\ \theta_1 \end{array}\right]$$

- And empirical loss will be

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} (y_i - (\theta_0 + \theta_1 x_i))^2$$

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \{y_i - (\left[\begin{array}{cc} 1 & x_i \end{array}\right] . \left[\begin{array}{c} \theta_0 \\ \theta_1 \end{array}\right])\}^2$$

Adding Features

- In K dimensions the regression function f(x) we estimate will look like

$$f(X) = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + ... + \theta_K x_{iK}$$

- So the empirical loss would

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\{y_i - (\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + ... + \theta_K x_{iK})\}^2$$

- Representing with matrices

$$J(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\{y_i - (\begin{bmatrix} 1 & x_{i1} & x_{i2} & \dots & x_{iK} \end{bmatrix} . \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ . \\ . \\ . \\ \theta_K \end{bmatrix})\}^2$$

Empirical Loss with K Features and N Data Points in Matrix Representation

- Representing empirical loss in Matrix form

$$\mathrm{J}(\theta) = \frac{1}{2N} \left\| \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ Y_N \end{bmatrix}}_{\mathbf{Y}} - \underbrace{\begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1K} \\ 1 & x_{21} & x_{22} & \dots & x_{2K} \\ & & \dots & & \\ & & \dots & & \\ & & \dots & & \\ 1 & x_{N1} & x_{N2} & \dots & x_{NK} \end{bmatrix}}_{\mathbf{X}} . \underbrace{\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ . \\ . \\ . \\ \theta_K \end{bmatrix}}_{\boldsymbol{\theta}} \right\|^2$$

Solve by Setting Partial Derivatives to Zero

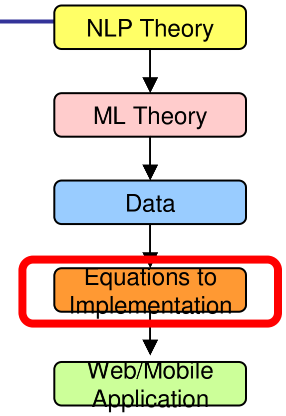
- Remember, to find the minimum empirical loss we set the partial derivatives to zero
- We can still do the same in matrix form, we have to set the derivatives to zero

$$\begin{aligned}\frac{\partial J(\theta)}{\partial \theta} &= \frac{1}{2N}\frac{\partial}{\partial \theta}||Y - X\theta||^2 \\ &= \frac{1}{2N}\frac{\partial}{\partial \theta}(Y - X\theta)^T(Y - X\theta) = 0\end{aligned}$$

- Solving the above equation we get our best set of parameters

$$\theta^* = (X^T X)^{-1} X^T Y$$

Equations to Implementation



$$\theta^* = (X^T X)^{-1} X^T Y$$

- Given out N training data points we can build X and Y matrix and perform the matrix operations
- Can use MATLAB
- Or write your own, Matrix multiplication implementation
- Get the theta matrix
- For any new test data plug in the x values (features) in our regression function with the best theta values we have

Back to Our Happiness Prediction Regression Model

- Xi1 represented count of +ve words
- (Xi1, Yi) pair were used to build simple linear regression model
- We added one more feature Xi2, representing count of –ve words
- (Xi1, Xi2, Yi) can be used to build multiple linear regression model

- Our training data would look like
  - (1, 3, 4)
  - (0, 6, 1.8)
  - .
  - .
  - .
  - (2, 0, 8.9)

(Xi1 , Xi2 , Yi)

- From this we can build **X** and **Y** Matrix and find the best theta values
- For N Data points, we will get Nx3 **X** matrix, Nx1 **Y** matrix and 3X1 **θ** matrix

More Features? Feature Engineering

- So far we have only two features, is it good enough?
- Should we add more features?
- What kind of features can we add?
  - Ratio of +ve/-ve words
  - Normalized count of +ve words
  - Is there a verb in the sentence?

- We need to think what are the kinds of information that may better estimate the Y values
- If we add above 3 features, what is the value of K?

Polynomial Regression

- For our Simple Happiness Scoring Model we tried to predict a linear function that fits the data points (x_i, y_i)

- What if linear function such as $f(x) = \theta_0 + \theta_1 x$ is not the right fit for data

- We can still fit a regression line but with 2^{nd} order polynomial

- Our function would look like

$$f(x) = \theta_0 + \theta_1 x^1 + \theta_2 x^2$$

Polynomial Regression

- To fit any m order polynomial on the given data we will follow the similar process as before of minimizing the empirical loss

- The regression function would be

$$f(x) = \theta_0 + \theta_1 x^1 + \theta_2 x^2 + \ldots + \theta_m x^m$$

- Parameters still given by the same solution

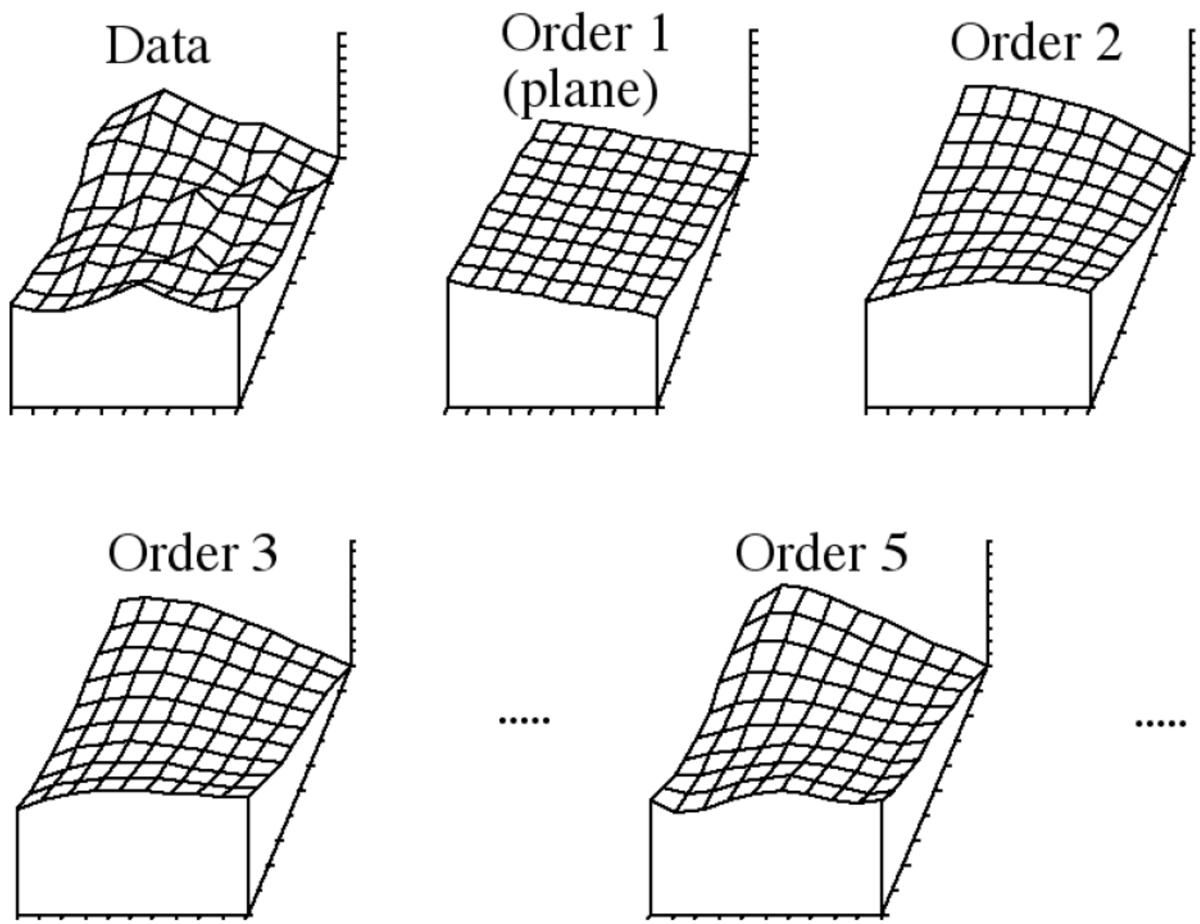
$$\theta^* = (X^T X)^{-1} X^T Y$$

- Solution is linear in parameters but nonlinear in the inputs

K Features, M Order Polynomial and N Data Points

- With K=1, we get a regression line, with K=2 we get a plane
- With M=1 we get a straight line or plane
- With M=2 we get a curved line or plane
- So with K=2 and M=2 ?

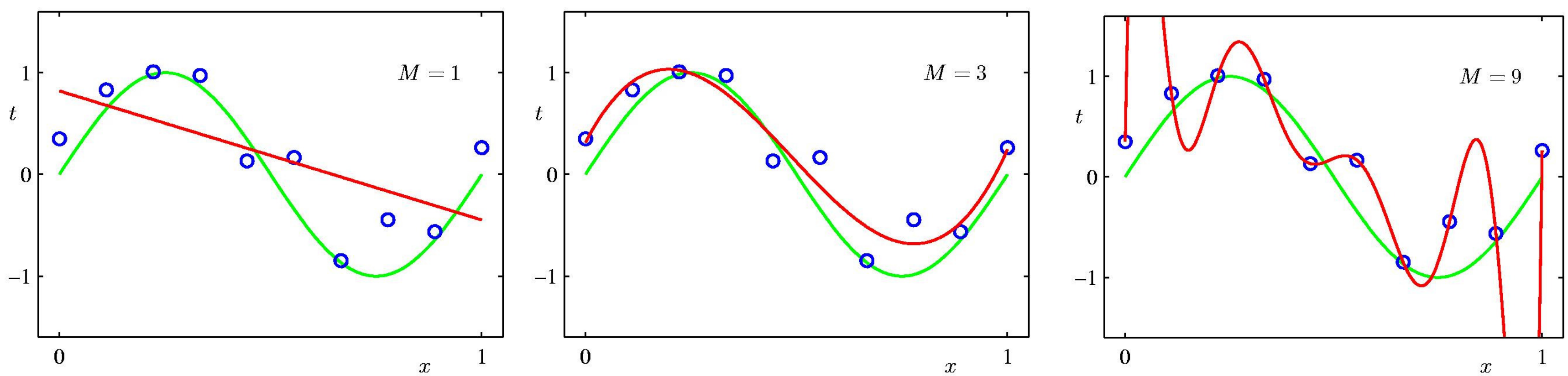
Trend Surface



Trend Surfaces for different orders of polynomial [1]

Overfitting

- Higher order of polynomial should be used with caution though
- Higher order polynomial can fit the training data too closely especially when few training points, with the generalization error being high
- Leave one out cross validation allows to estimate generalization error better
  - If N data points use N-1 data points to train and use 1 to test



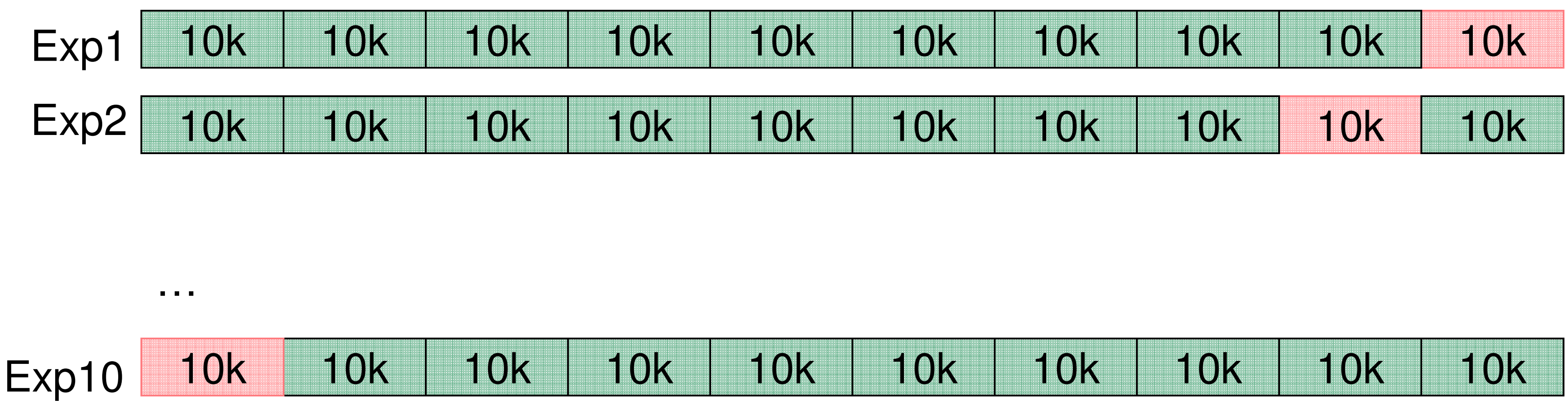
Higher order of polynomial overfitting with few data points [2]

Testing Our Model

- Our goal was to build the best statistical model that would automate the process of scoring a chunk of text (Happiness Score)
- How can we tell how good is our model?
- Remember previously we said let us assume we have 100,000 status updates
- Instead of using all 100K sentences let use the first 90K to build the model
- Use rest of 10K to test the model

10-fold Cross Validation

- 10 fold cross validation
  - We trained on first 90K (1 to 90,000)
  - Tested on (90,001 to 100,000)
  - But we can do this 10 times if we select different 10K of test data point each time

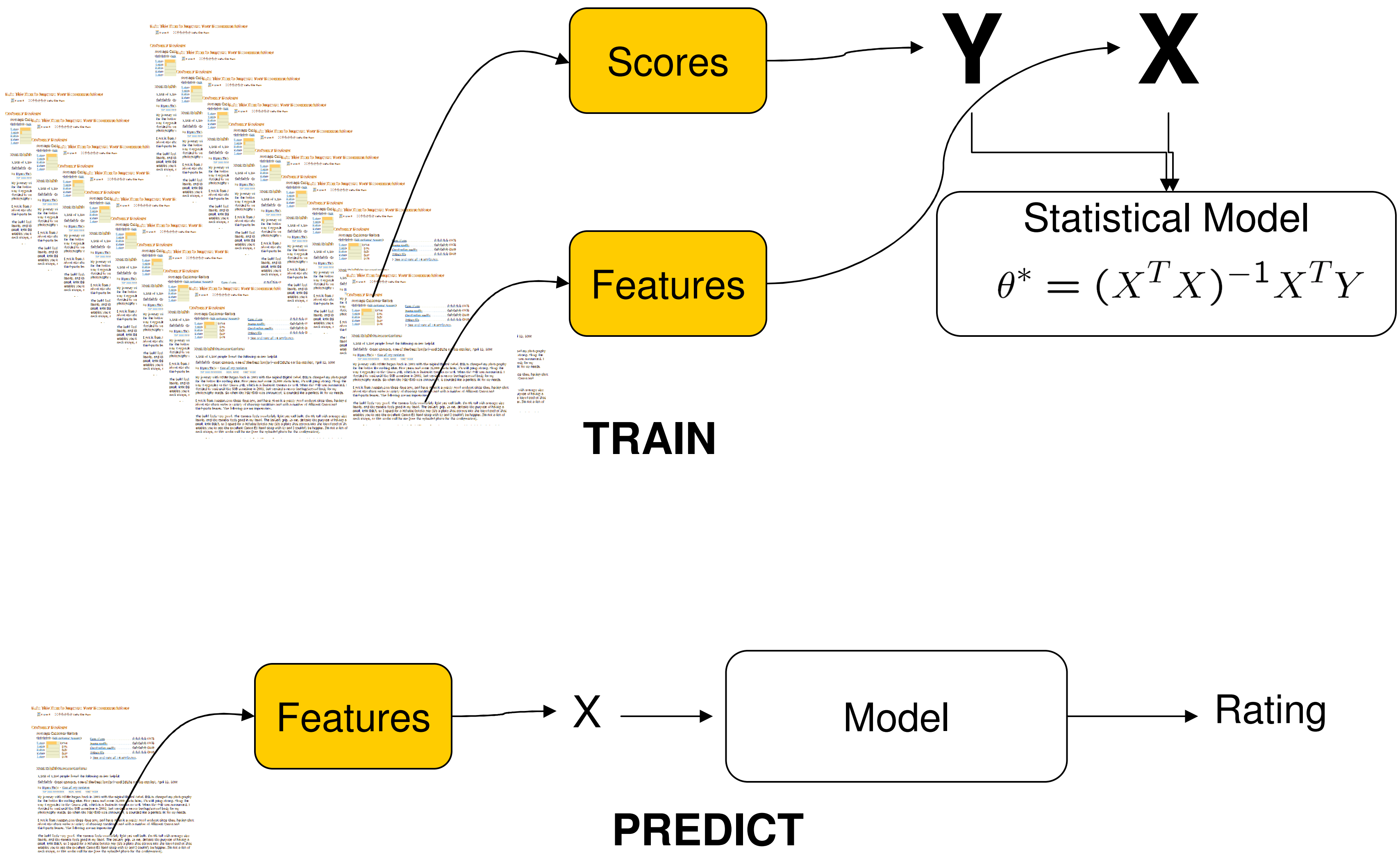


- 10 experiments, build model and test times with 10 different sets of training and test data
- Average the accuracy across 10 experiments
- We can do any N-fold cross validation to test our model

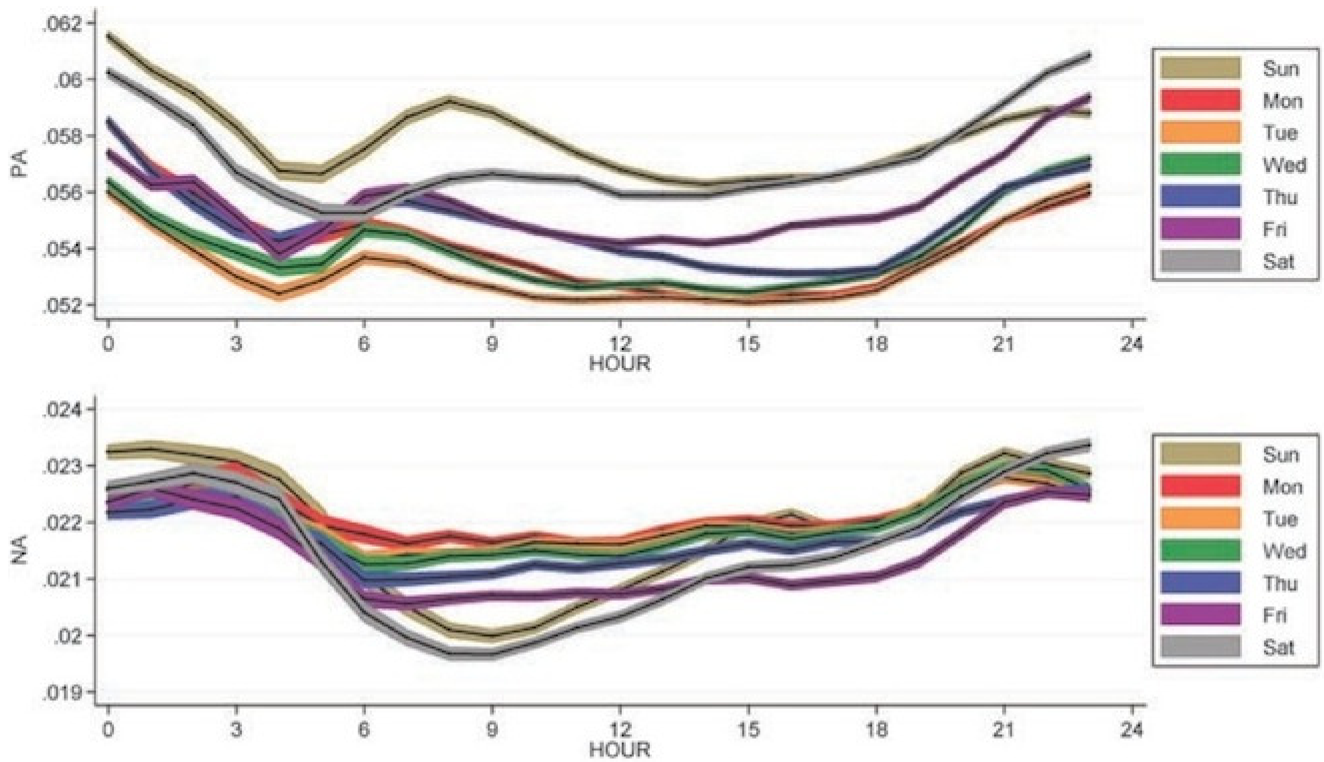
Scores from Text, What Else Can They Represent?

- Given a facebook status update we can predict happiness score
- But we can use the same modeling technique in many other problems
  - Summarization: Score may represent importance
  - Question Answering: Score may represent relevance
  - Information extraction : Score may represent relation
- We need to engineer features according to the problem
- Many uses of the statistical technique we learned today

Reviews to Automatic Ratings



Tweets to Mood Score



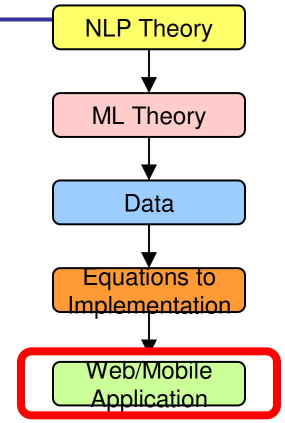
"Diurnal and Seasonal Mood Vary with Work, Sleep, and Daylength Across Diverse Cultures." By Scott A. Golder and Michael W. Macy. Science, Vol. 333, September 30, 2011.

- Can you now implement a regression model to predict Mood Score for new tweets?

Interesting Research on Twitter Data

- **Predict Elections :** Tumasjan, et. al, “Predicting elections with **twitter**: What 140 characters reveal about political sentiment,” AAAI 2010
- **Understand Mood Variations** : Golder & Macy, “Diurnal and Seasonal Mood Vary with Work, Sleep, and Daylength Across Diverse Cultures”, Science, 30 September 2011: Vol. 333 no. 6051 pp. 1878-1881
- **Find Influential People** : Weng et al, “Twitterrank: finding topic-sensitive influential twitterers” WSDM, 2010
- **Usage in Disease Outbreak** : Chew et al.,“Pandemics in the age of Twitter: content analysis of Tweets during the 2009 H1N1 outbreak,” PloS One, 5(11), 2010
- **Try predicting Stock Market** : Bollen et al, “Twitter Mood Predicts the Stock Market”, Journal of Computational Science, Mar, 2011

What Else Can We Do?



- Sentiment Analysis
- Information Extraction
- Question Answering
- Search
- Text Mining
- Story Tracking
- Summary Generation
- Event Detection
- Entity Extraction
- .
- .
- .
- many more

Think about the Final Project!

Finding Partner for the Final Project

- You can do the project alone!
  - Some students prefer this
- You can team up with 1 more person
  - 2 person team needs to present a project that require 2 person effort
- Go to courseworks and start posting what you are interested in and find a partner

Example Final Project

Sentiment on various topics

- Automatically detect the sentiment of a person on a range of topics and update the model whenever new information is provided

Readings

- 1.1, 3.1, 4.1 Bishop Book
- 23.1.1, 23.1.2, 23.1.3 Jurafsky & Martin Book

References

- [1] http://biol09.biol.umontreal.ca/PLcourses/Polynomial_regression.pdf
- [2] Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2006
- [3] Hastie, Tibshirani and Friedman, Elements of Statistical Learning, 2001