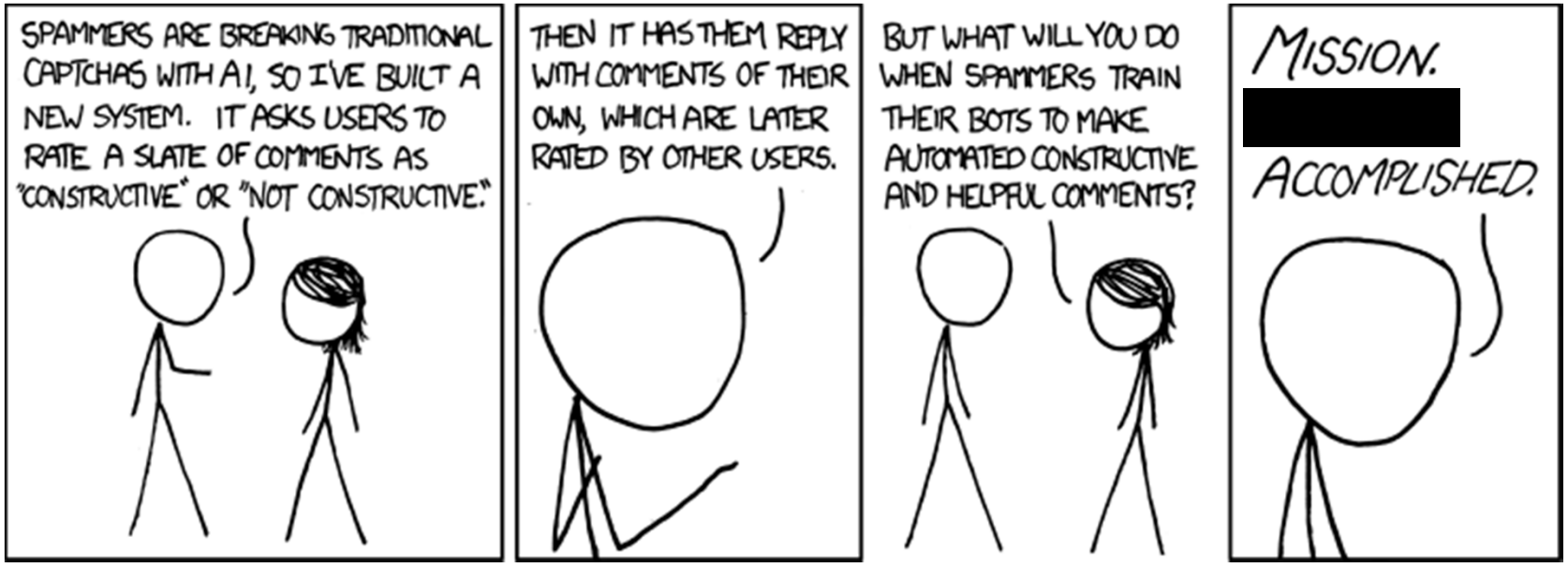
UNIVERSITY of WASHINGTON

Data Science UW
Methods for Data Analysis

Intro to Natural Language Processing
Lecture 10
Steve Elston


SPAMMERS ARE BREAKING TRADITIONAL CAPTCHAS WITH AI, SO I'VE BUILT A NEW SYSTEM. IT ASKS USERS TO RATE A SLATE OF COMMENTS AS "CONSTRUCTIVE" OR "NOT CONSTRUCTIVE."
THEN IT HAS THEM REPLY WITH COMMENTS OF THEIR OWN, WHICH ARE LATER RATED BY OTHER USERS.
BUT WHAT WILL YOU DO WHEN SPAMMERS TRAIN THEIR BOTS TO MAKE AUTOMATED CONSTRUCTIVE AND HELPFUL COMMENTS?
MISSION. [redacted] ACCOMPLISHED.

Topics

- Review
- Text normalization
- Term document matrix
- Text classification
- Topic models - Latent Dirichlet Allocation

Review

- Bayes models
  - Hierarchical models
  - Bayesian model selection – Bayes factor
  - Bayesian hypothesis testing
- MCMC diagnostics
- Naive Bayes models

Bayesian Model Summary

- Bayesian view of the world includes updating/changing beliefs new observations
- Bayesian view takes prior beliefs into account
- Based on Bayes theorem

$$P(A|B) = P(B|A)\frac{P(A)}{P(B)}$$

- Can use simplified formulation with no P(B)

$$P(A|B) \propto P(B|A)P(A)$$

Posterior Distribution

The Likelihood

Prior Distribution

Multi-level or Hierarchical Bayes Model

Extending Bayesian model

> Bayes rule becomes

$$P(\theta, \omega | D) = P(D | \theta, \omega)\, p(\theta, \omega)$$

$$= P(D | \theta)\, p(\theta | \omega)\, p(\omega)$$

Posterior Distribution

The Likelihood

Prior Distribution of θ given ω

Prior Distribution of ω

> Example: for beta prior and Bernoulli likelihood:

Prior of $\omega = \text{Beta}(A_\omega, B_\omega)$

$$P(\theta, \omega | D) = \text{Bernoulli}(\theta)\, \text{Beta}(\omega (K-2) + 1, (1 - \omega)(K - \quad + 1)$$

Joint Prior

Hypothesis Testing with Bayes Models

Use HCI to perform hypothesis tests

- Analogous to hypothesis tests on bootstrap resampled distributions
- Test conditions for **posterior** distribution
  - If HDI overlap; accept Null Hypothesis
  - If no HDI overlap reject Null Hypothesis
- HDI is different from Confidence Interval
  - HDI is for interval with greatest probability mass
  - Difference with CI is greatest for asymmetric prior
- Tests can be one-sided or two-sided

Diagnostics for MCMC

Multiple ways to look at convergence

- Summary statistics
  - Mean, median, se, time series se, quantiles
  - Plot cumulative mean and quantiles
  - Plot trace of each chain
  - Plot posterior distribution
- Plots based on convergence of multiple chains
  - Gelman-Rudin plot of chain convergence
  - Compares shrinkage of between chain and within chain variance
  - Should converge to 1.0

Diagnostics for MCMC

Detect convergence issues

- High rejection rate inhibits convergence
- High autocorrelation inhibits convergence
- Use ACF
- Effective Sample Size

$$ESS = N / \left(1 + 2 \sum_K ACF(k)\right)$$

Naïve Bayes

Simplify the conditional probability calculation

> With $\{x_1, x_2, \ldots, x_n\}$ independent:

$$p(x_i | x_{i+1}, \ldots, x_n, C_k) = p(x_i, | C_k)$$

> The probability of class C_k is the joint distribution:

$$p(C_k | x_1, x_2, \ldots, x_n) \propto p(C_k) \prod^{N}_{j=1} p(x_j | C_k)$$

> And the most likely class y_{hat} is:

$$y_{hat} = \text{argmax}_k \left[p(C_k) \prod^{N}_{j=1} p(x_j | C_k) \right]$$

No Prior

Naïve Bayes Classifiers

Different distributions lead to different classifiers

- Difference Naïve Bayes models are not the same!
- Normal naïve Bayes classifier
- Multinomial naïve Bayes classifier

$$\text{Log}(p(C_k \mid x)) \propto \log[\, p(C_k) \prod^{N}_{j=1} p_{kj}^{\,Xi} \,]$$

$$= \log(\, p(C_k) \,) + \sum^{N}_{j=1} xi \log(\, p_{kj} \,)$$

- Bernoulli naïve Bayes classifier

$$p(x \mid C_k) = \prod^{N}_{j=1} p_{kj}^{\,Xi} (1 - p_{kj})^{(1 - Xi)}$$

Text Data are Everywhere!

- Most of the world's data is unstructured:
  - 2009 HP Survey: 70%
  - Gartner: 80%
  - Teradata: 85%
  - **Beware of industry estimates!!**
- How much data?
  - Twitter has more text data recorded than all that has been written in print in the history of mankind. (http://www.internetlivestats.com/twitter-statistics/)

Many Applications of Text Analytics

- Intelligent applications
  - Assistants
  - Chat bots
- Classification
  - Sentiment analysis
  - SPAM detection
- Speech recognition
- Search
- Information retrieval

How can we analyze text data?

Need to transform to a structured form

- Organize text documents into corpus
- Normalize the text to remove unneeded content
- Tokenize text
  - Words
  - N-grams
  - Sentences
- Analytics models

Methods of Text Analysis

Broad and deep field

- Bag of words model
  - Widely used
  - Based on word frequency
  - Assumes **exchangeability** of words
  - Use term-document or document-term matrix
- Classification
  - Term frequency as features
- Part of Speech (PoS) Tagging
  - Annotate corpus
  - Create tree of PoS

Methods of Analysis

Wide range of models

- Latent Sematic Analysis (LSA)
  - Which documents are closely related?
- Topic models
  - Allocate the probability a document contains a topic
  - Latent Dirichlet Allocation (LDA)
- Clustering
  - Use text distance
  - K-means
  - Hieratical
- Named entity identification

Text Normalization

Need text in uniform format suitable for application

- Remove extraneous symbols
  - White space
  - Punctuation
  - Numbers
  - Etc.
- Convert to lower cases
- Remove extraneous words (tokens)
  - Frequent but useless words = stop words

Text Normalization

Need text in uniform format suitable for application

- Stem words to root
  - **stemming** is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form
  - E.g. verbs in different tense are same word
  - Pioneered by Julie Beth Lovens (1968)
  - Porter (1980, 2000) is common algorithm for English
  - e.g. relies, relied, rely = reli
- Substitute synonyms

Text Normalization (Pre processing)

- Strip extra white space:

  I <3 statistics, it's my \u1072 fAvoRitE!! 11!!!

  ↓

  I <3 statistics, it's my \u1072 fAvoRitE!! 11!!!

- Remove Unicode text

  I <3 statistics, it's my \u1072 fAvoRitE!! 11!!!

  ↓

  I <3 statistics, it's my fAvoRitE!! 11!!!

- Lower case

I <3 statistics, it's my fAvoRitE!! 11!!! ⟶ i <3 statistics, it's my favorite!! 11!!!

- Remove punctuation

i <3 statistics, it's my favorite!! 11!!! ⟶ i 3 statistics its my favorite 11

Text Normalization (Pre processing)

- Remove numbers

  i 3 statistics its my favorite 11 → i statistics its my favorite

- Remove stop words

  i statistics its my favorite → statistics favorite

- Stem words (optional)

  statistics favorite → statisti favori

- R-demo

Term Document Matrix

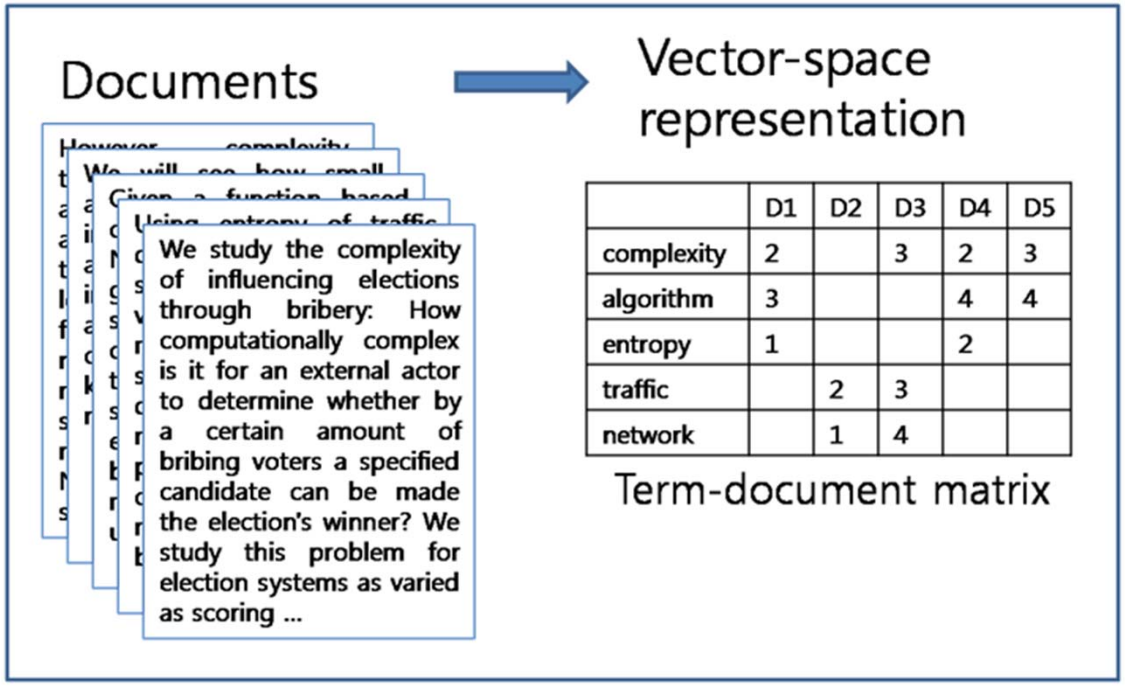
Representation of **Bag of Words model**

- Terms in rows documents in columns
- Document term matrix is transpose
- Using the distribution of a document's TF (TF-IDF) values.
  - Characterize writing styles
  - Comparing authors
  - Determining original authors
  - Finding plagiarism

Term Document Matrix

Representation of **Bag of Words model**

> **Term Frequency (TF)** weighting is the count of term in each document



	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

Term-document matrix

> Generally very sparse matrix

Term Document Matrix

Representation of **Bag of Words model**

- Terms in rows documents in columns
- Document term matrix is transpose
- **Term Frequency (TF)** weighting is the count of term in each document
- **Inverse Document Frequency (TFIDF)** weighting accounts for few documents containing term

$$IDF = \log\left(\frac{\#Documents}{\#Documents\ with\ Word}\right)$$

Term Document Matrix

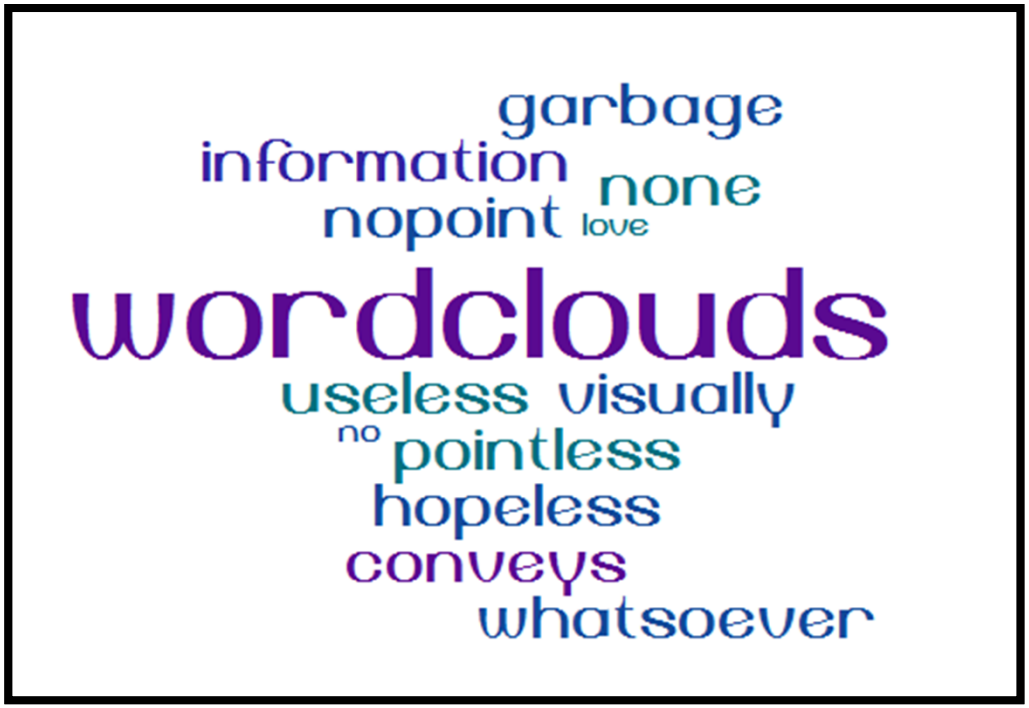
Reweight TF by IDF = TFIDF matrix

> Can prevent few documents with frequent terms from dominating.

$$TF - IDF = \log\left(\frac{\#Documents}{\#Documents\ with\ Word}\right) \times f(Word)$$

Wordclouds

Completely useless display of information that people love to see.



Sentiment Analysis

Document classification problem

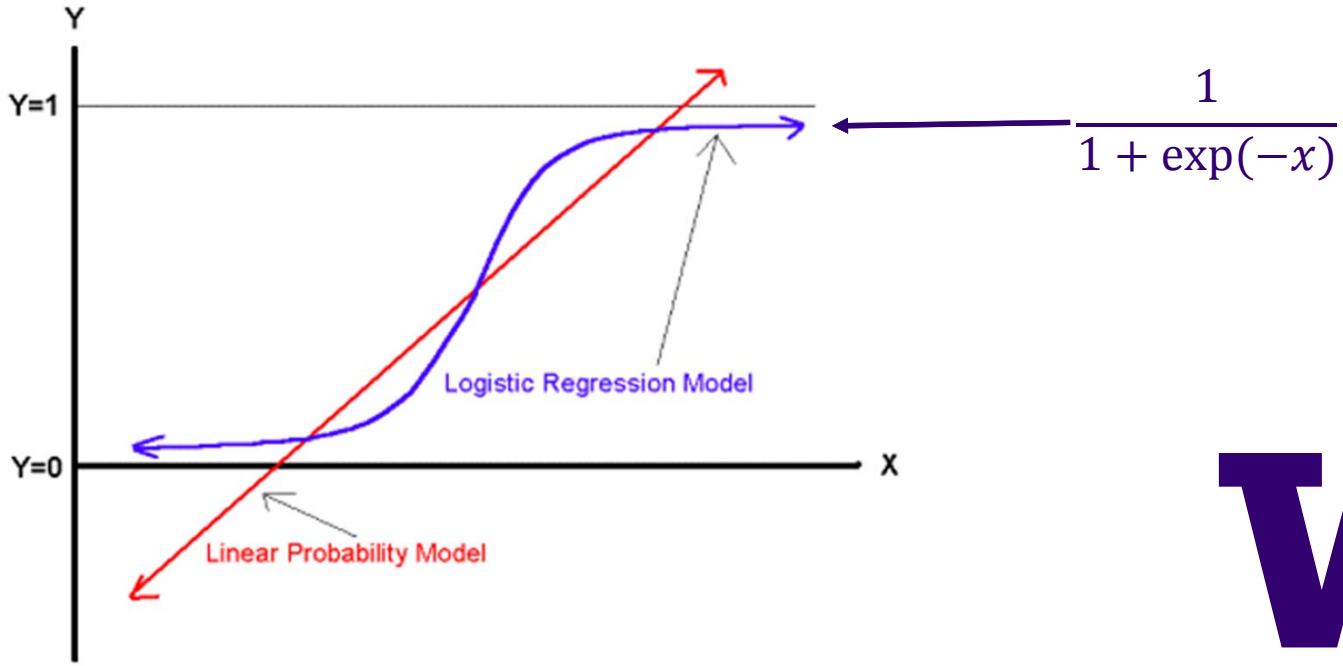
- Use TDM or TFIDF weighted TDM as features
- Use marked cases for training and evaluation of model
- Sparse matrix requires regularization
  - Feature selection
  - SVD/PCA
  - Ridge and Lasso methods - elasticnet

Recall: Logistic Regression

$$p_i = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1))}$$

> As $(\beta_0 + \beta_1 x_1)$ gets really big, p approaches 1.

> As $(\beta_0 + \beta_1 x_1)$ gets really small, p approaches 0.



Recall: Ridge and Lasso Regression

- Ridge regression limits influence of each feature on model
- Minimizes the least squares of the error plus a regularization term that is a product of a constant and the sum of squared coefficients :

$$\min \sum (y - y_i)^2 + \alpha \sum \beta^2$$

- Essentially this is preventing the coefficients from getting too large.
- Lasso regression minimizes the same with the addition of a 'regularization' term:

$$\min \sum \left(y - y_j\right)^2 \qquad \text{Such that} \qquad \sum |\beta_i| < \lambda$$

- Limits absolute sum of coefficients
- Combination gives **elasticnet**

Metrics for Classification

Confusion matrix

	Predicted Positive	Predicted Negative
Actual Positive	**TP**	**FN**
Actual Negative	**FP**	**TN**

Metrics for Classification

- Accuracy = TP + TN / (TP + TN + FP + FN)
- Precision or positive predictive value = TP/(TP + FP)
- Recall = TP/(TP + FN)
- + Many others!

Topic Models

How do we allocate documents to topics?

- Unsupervised learning problem
- Latent Dirichlet Allocation + others
- Bayesian model

Topic Models

Latent Dirichlet Allocation model

- Fixed number of (sub) topics, k
- Find probability of document containing topic
- Only known word frequencies for documents in corpus
- All other variables are estimated or **latent**

Overview of Latent Dirichlet Allocation Model

Dirichlet distribution is conjugate of multinomial and categorical distribution

All we actually know:

ω_{ij} is a specific word in document I

What we want to know (latent):

θ_i is the topic distribution of document i

We also need to estimate (latent):

Φ_k is the word distribution for topic k

z_{ij} is the topic of the jth word in document i

The Model and Its Priors

Multinomial model

$z_{ij} \sim \text{multinomiall}(\theta_i)$

$\omega_{ij} \sim \text{multinomial}(\Phi_k)$

With Dirichlet priors with parameters α and β:

$\theta_i \sim \text{Dir}(\alpha)$

$\Phi_k \sim \text{Dir}(\beta)$

Generally use uniform priors across topics

Likelihood from TD matrix

Summary

- Text normalization
- Term document matrix
  - Bag of words model
- Text classification
  - Many applications
  - TDM as features
- Topic models - Latent Dirichlet Allocation
  - Allocate topics to documents
  - Unsupervised learning

The END!

Please complete a course review

Thank you!!!!