Practical Methods for the Analysis of "Big Data"

Module 3: Text Classification and Topic Models

Philip A. Schrodt

The Pennsylvania State University
schrodt@psu.edu

Workshop at the Odum Institute
University of North Carolina, Chapel Hill
20-21 May 2013

Topics: Module 3

Distinctions between statistical and machine learning approaches

Metrics in machine learning

Text as a statistical object

- Support Vector Machines

Topic modeling: Latent Dirichlet Allocation

Sequence models

- Levenshtein distance
- HMM
- CRF
- Additional comments on sequences

Followups from Monday

- Machine learning implemented on Hadoop/Amazon Elastic MapReduce (EMR) : https://cwiki.apache.org/confluence/display/MAHOUT/Algorithms
- Yet another list of open source machine learning projects: http://mloss.org/software/
- A visualization using Twitter data: http://posterhall.org/igert2013/posters/425

Regular Expression

"Calculus for string manipulation"

Regular Expr. Recipes
Regex Recipes for Windows

Regex Reference
Basic Regex Syntax
Advanced Regex Syntax
Unicode-Specific Syntax
Flavor-Specific Syntax
Flavor Comparison
Replacement Syntax

More Information
Introduction
Quick Start
Tutorial
Tools and Languages
Examples
Books
Reference
Print PDF
About This Site
RSS Feed & Blog

Regular Expression Basic Syntax Reference

Characters		
Character	**Description**	**Example**
Any character except `[\^$.|?*+()`	All characters except the listed special characters match a single instance of themselves. { and } are literal characters, unless they're part of a valid regular expression token (e.g. the {n} quantifier).	a matches a
\ (backslash) followed by any of `[\^$.|?*+(){}`	A backslash escapes special characters to suppress their special meaning.	\+ matches +
`\Q...\E`	Matches the characters between \Q and \E literally, suppressing the meaning of special characters.	\Q+-*/\E matches +-*/
`\xFF` where FF are 2 hexadecimal digits	Matches the character with the specified ASCII/ANSI value, which depends on the code page used. Can be used in character classes.	\xA9 matches © when using the Latin-1 code page.
`\n`, `\r` and `\t`	Match an LF character, CR character and a tab character respectively. Can be used in character classes.	\r\n matches a DOS/Windows CRLF line break.
`\a`, `\e`, `\f` and `\v`	Match a bell character (\x07), escape character (\x1B), form feed (\x0C) and vertical tab (\x0B) respectively. Can be used in character classes.	
`\cA` through `\cZ`	Match an ASCII character Control+A through Control+Z, equivalent to \x01 through \x1A. Can be used in character classes.	\cM\cJ matches a DOS/Windows CRLF line break.
Character Classes or Character Sets [abc]		
Character	**Description**	**Example**
[(opening square bracket)	Starts a character class. A character class matches a single character out of all the possibilities offered by the character class. Inside a character class, different rules apply. The rules in this section are only valid inside character classes. The rules outside this section are not valid in character classes, except for a few character escapes that are indicated with "can be used inside character classes".	
Any character except	All characters except the listed special characters	[abc] matches a, b or c

Source: http://www.regular-expressions.info/reference.html

Stanford

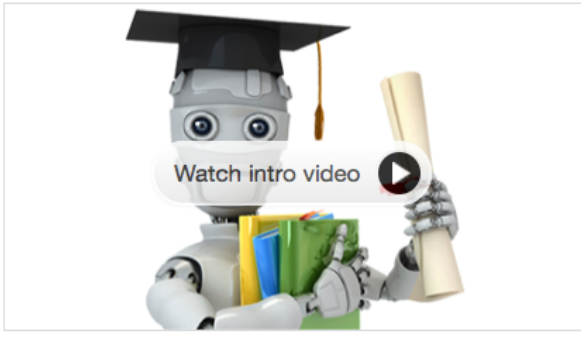
Machine Learning

Andrew Ng

Learn about the most effective machine learning techniques, and gain practice implementing them and getting them to work for yourself.

Workload: 5-7 hours/week

Preview



Sessions:

Apr 22nd 2013 (10 weeks long) Sign Up

Future sessions Add to Watchlist

About the Course

Machine learning is the science of getting computers to act without being explicitly programmed. In the past decade, machine learning has given us self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome. Machine learning is so pervasive today that you probably use it dozens of times a day without knowing it. Many researchers also think it is the best way to make progress towards human-level AI. In this class, you will learn about the most effective machine learning techniques, and gain practice implementing them and getting them to work for yourself. More importantly, you'll learn about not only the theoretical underpinnings of learning, but also gain the practical know-how needed to quickly and powerfully apply these techniques to new problems. Finally, you'll learn about some of Silicon Valley's best practices in innovation as it pertains to machine learning and AI.

About the Instructor

Andrew Ng
Stanford University

Introduction to Artificial Intelligence

UDACITY

Intermediate

INSTRUCTORS

Sebastian Thrun

Peter Norvig

Take the Class

Class Summary

The objective of this class is to teach you modern AI. You will learn about the basic techniques and tricks of the trade. We also aspire to excite you about the field of AI.

What Should I Know?

Some of the topics in Introduction to Artificial Intelligence will build on probability theory and linear algebra. You should have understanding of probability theory comparable to that at our ST101: Introduction to Statistics class

What Will I Learn?

This class introduces students to the basics of Artificial Intelligence, which includes machine learning, probabilistic reasoning, robotics, and natural language processing.

Syllabus

Overview of AI

Statistics, Uncertainty, and Bayes networks

Machine Learning

Logic and Planning

Markov Decision Processes and Reinforcement Learning

Hidden Markov Models and Filters

Adversarial and Advanced Planning

Image Processing and Computer Vision

Robotics and robot motion planning

Natural Language Processing and Information Retrieval

Course Instructors

Sebastian Thrun

Instructor

Sebastian Thrun is a Research Professor of Computer Science at Stanford University, a Google Fellow, a member of the National Academy of Engineering and the German Academy of Sciences. Thrun is best known for his research in robotics and machine learning, specifically his work with self-driving cars.

Peter Norvig

Instructor

Topics: Module 3

Distinctions between statistical and machine learning approaches

Metrics in machine learning

Text as a statistical object
- Support Vector Machines

Topic modeling: Latent Dirichlet Allocation

Sequence models
- Levenshtein distance
- HMM
- CRF
- Additional comments on sequences

Methods of Modeling

Classical ("frequentist") statistics

- Objective is determining whether a variable has a non-zero effect: "significance tests"
- Effective in experimental and other randomized settings, but generally useless for predictive models

Bayesian statistics

- Objective is determining a probability of the impact of a variable based on your prior knowledge and the new data
- Corresponds to how most people actually think about data but has only become computationally feasible in the past twenty years

Machine Learning

- Very flexible methods of determining relationships
- Robust with respect to loosely structured data
- Problem: No [widely accepted] theory of error

Distinctions between statistical and machine learning approaches

- Focus on out-of-sample validation, not standard error of coefficients
- Collinearity is an asset, not a liability
- Assumption—and exploitation—of heterogeneous subpopulations
- Danger of overfitting
- Missing values can be data
- Sparse datasets: most indicators are not measured on most case
- Non-linear, and consequently the cases » variables constraint need not apply
- Diffuse knowledge structures
- ML methods are frequently just the application of a “common sense” algorithm, whereas statistical approaches often require detailed derivations in the

Prediction vs frequentist significance tests

- Significance becomes irrelevant in really large data sets: true correlations are almost never zero
- Emphasis is on finding reproducible patterns, but in any number of different frameworks
- Testing is almost universally out-of-sample
- Some machine learning methods are explicitly probabilistic—though usually Bayesian—others are not
- Values of individual coefficients are usually of little interest because there are so many of them and they are affected by collinearity

Topics: Module 3

- Distinctions between statistical and machine learning approaches
- Metrics in machine learning
- Text as a statistical object
  - Support Vector Machines
- Topic modeling: Latent Dirichlet Allocation
- Sequence models
  - Levenshtein distance
  - HMM
  - CRF
  - Additional comments on sequences

Classification Matrix

Relationships among terms

		Condition (as determined by "Gold standard")		
		Condition Positive	Condition Negative	
Test Outcome	Test Outcome Positive	**True Positive**	**False Positive** (Type I error)	Positive predictive value = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Test Outcome Positive}}$
	Test Outcome Negative	**False Negative** (Type II error)	**True Negative**	Negative predictive value = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Test Outcome Negative}}$
		Sensitivity = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Condition Positive}}$	**Specificity** = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Condition Negative}}$	

Accuracy, precision and recall

Precision and recall are then defined as:[4]

$\text{Precision} = \frac{tp}{tp+fp}$

$\text{Recall} = \frac{tp}{tp+fn}$

Recall in this context is also referred to as the True Positive Rate or Sensitivity, and precision is also referred to as Positive predictive value (PPV); other related measures used in classification include True Negative Rate and Accuracy. True Negative Rate is also called Specificity.

$\text{True negative rate} = \frac{tn}{tn+fp}$

$\text{Accuracy} = \frac{tp+tn}{tp+tn+fp+fn}$

Source: http://en.wikipedia.org/wiki/Precision_and_recall

F1 score

The traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall:

$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$.

The general formula for positive real β is:

$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$.

The formula in terms of Type I and type II errors:

$F_\beta = \frac{(1+\beta^2) \cdot \text{true positive}}{((1+\beta^2) \cdot \text{true positive} + \beta^2 \cdot \text{false negative} + \text{false positive})}$

Two other commonly used F measures are the F_2 measure, which weights recall higher than precision, and the $F_{0.5}$ measure, which puts more emphasis on precision than recall.

The F-measure was derived so that F_β "measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision". It is based on van Rijsbergen's effectiveness measure

$E = 1 - \left(\frac{\alpha}{P} + \frac{1-\alpha}{R}\right)^{-1}$.

Their relationship is $F_\beta = 1 - E$ where $\alpha = \frac{1}{1+\beta^2}$.

Source: http://en.wikipedia.org/wiki/F1_score

Metrics: Example 1

Table 3: Classification Table: REBELL

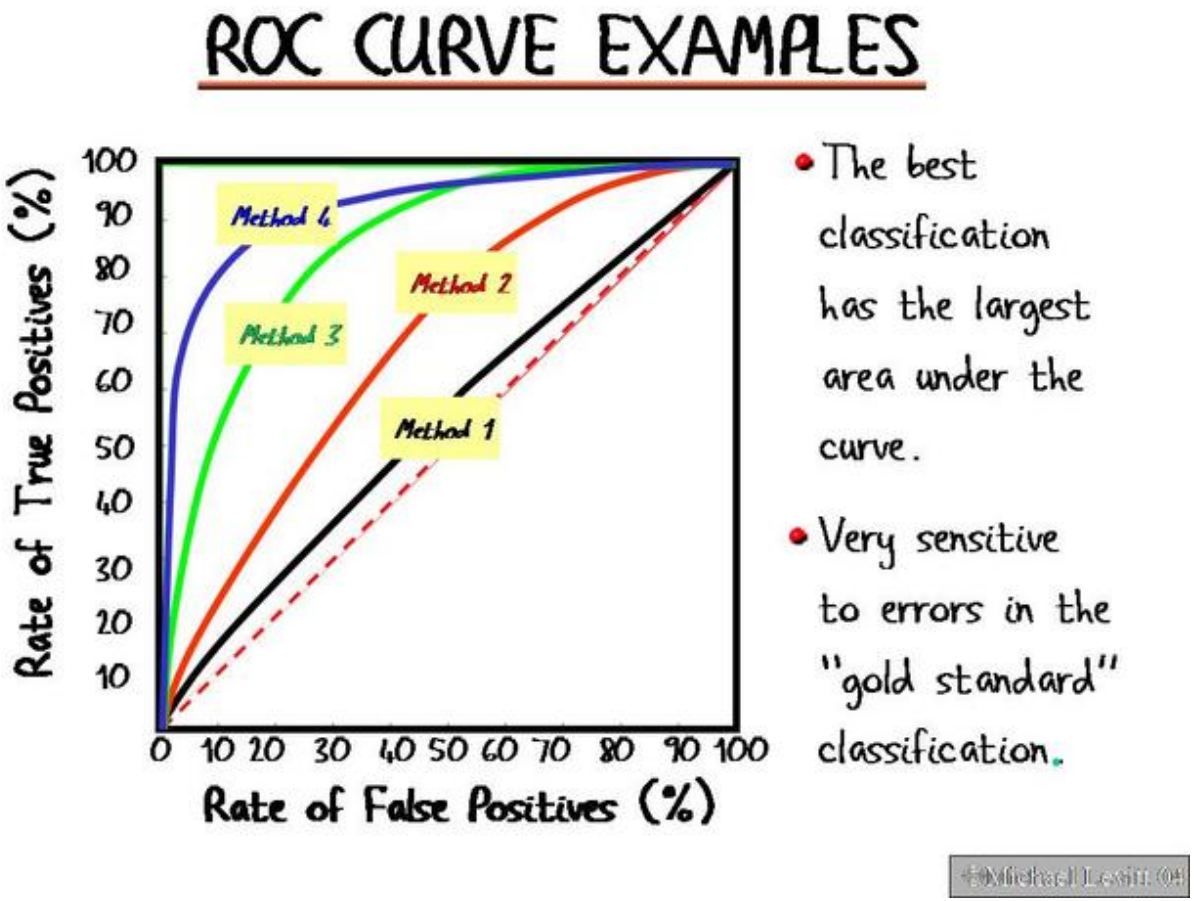
		true		
pred	0	1	row N	
0	904	592	1496	
1	126	283	409	
col N	1030	875	1905	
	Acc	0.623	AUC	0.732
	Spec	0.604	Sens	0.691
	Prec	0.323	F1	0.421
sLDA AUC		0.527		
ICEWS Reference Model				
	Acc	0.852		
	Spec	0.996		
	Sens	0.387		
	N	4437		

Metrics: Example 2

Middle East: Results

	Accuracy	Specificity	Sensitivity	AUC
		3-month		
ISR→PSE				
Mean	0.646	0.696	0.595	0.708
StDev	0.031	0.058	0.0612	0.025
PSE→ISR				
Mean	0.710	0.734	0.686	0.778
StDev	0.025	0.048	0.041	0.021
ISR→LBN				
Mean	0.639	0.691	0.587	0.683
StDev	0.029	0.078	0.062	0.031
LBN→ISR				
Mean	0.624	0.831	0.368	0.673
StDev	0.016	0.023	0.038	0.016

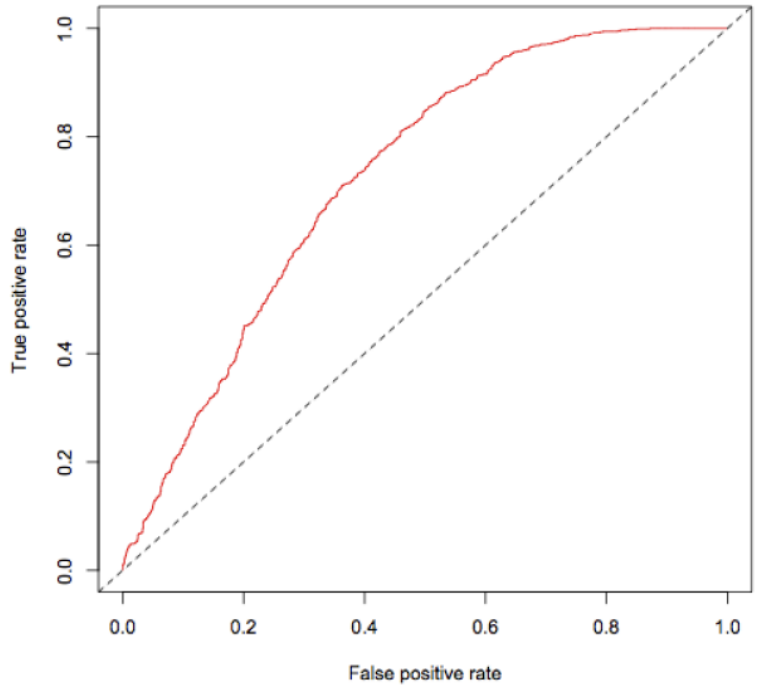
ROC Curve



Source: http://csb.stanford.edu/class/public/lectures/lec4/Lecture6/Data_Visualization/images/Roc_Curve_Examples.jpg

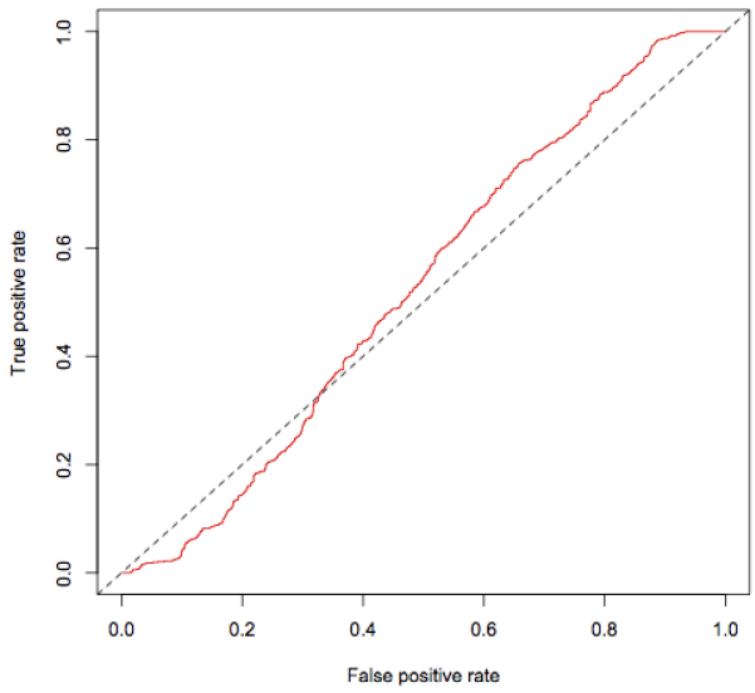
ROC Curve

LDA ROC: REBELL



ROC Curve

sLDA ROC: REBELL



Separation plots

TABLE 4 Rearrangement (and Coloring) of the Data Presented in Table 1 for Use in the Separation Plot

Country	Fitted Value ($\hat{p}$)	Actual Outcome (y)
B	0.364	0
F	0.422	1
D	0.728	0
A	0.774	0
E	0.961	1
C	0.997	1

FIGURE 2 Separation Plot Representing the Data Presented in Table 1

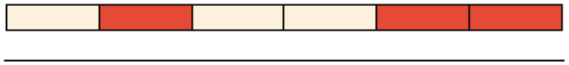
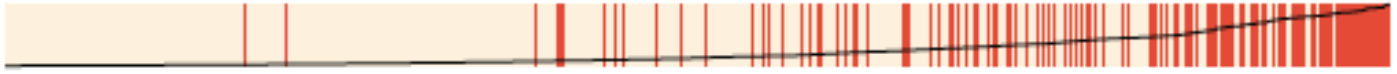


FIGURE 3 Separation Plot for a Larger Data Set

FIGURE 4 Adding a Graph of $\hat{p}$ to the Separation Plot



The Separation Plot: A New Visual Method for Evaluating the Fit of Binary Models

Brian Greenhill[1], Michael D. Ward[2], Audrey Sacks[3]

Article first published online: 20 JUN 2011

DOI: 10.1111/j.1540-5907.2011.00525.x

Issue

American Journal of Political Science

Volume 55, Issue 4, pages 991–1002, October 2011

Topics: Module 3

- Distinctions between statistical and machine learning approaches
- Metrics in machine learning
- Text as a statistical object
  - Support Vector Machines
- Topic modeling: Latent Dirichlet Allocation
- Sequence models
  - Levenshtein distance
  - HMM
  - CRF
  - Additional comments on sequences

What does a document look like as a statistical object?

- Mathematically:
  - it is a high-dimensional, sparse feature vector where the elements of the vector are the frequencies of specific words and phrases in the document
- Geometrically:
  - it is a point in a high-dimensional space

Therefore, anything you can do with points, you can do with documents

Term-Document matrix

	T1	T2	T3	T4	T5	T6	T7	T8
Doc1	2	0	4	3	0	1	0	2
Doc2	0	2	4	0	2	3	0	0
Doc3	4	0	1	3	0	1	0	1
Doc4	0	1	0	2	0	0	1	0
Doc5	0	0	2	0	0	4	0	0
Doc6	1	1	0	2	0	1	1	3
Doc7	2	1	3	4	0	2	0	2

Reduction of Dimensionality

- Computational incentives
  - Eliminate information that does not distinguish between documents: stop words
  - Combine words that have the same information: stemming
- Conceptual incentives
  - Deductive: identify groups of words or phrases that are consistently associated with the concepts you are trying to code
  - Inductive: give a set of related texts, find the common language, which may not be obvious
- Statistical incentives
  - Words that occur everywhere are noise and may make documents seem more similar than they are
  - Words that almost never occur are not useful for machine learning, even if they are very meaningful for a human coder

Zipf's Law (a.k.a. rank-size law)

The frequency of the occurrence of a word in a natural language is inversely proportional to its rank in frequency

In mathematics: $f_i \propto 1/r_i$

In English (or any other natural language): A small number of words account for most of word usage

Zipf's Law collides with statistical efficiency

Information theory: the information contained in an item of data is proportional to $log(f_i)$

Statistical Efficiency: the standard error of a parameter estimate is inversely proportional to the square root of the sample size

Upshot: Any content analysis must balance the high level of information contained in low-frequency words with the requirements of getting a sample of those words sufficiently large for reasonable parameter estimation

Statistical Methods I: Reduction of dimensionality

Objective: Approximate the high dimensional space with a space of lower dimensionality while preserving as much of the variance as possible in the original space.

- Factor analysis: correlation metric
- Principal components: Euclidean metric
- Correspondence analysis: chi-square metric

Result: Document can be characterized by a small number of composite indicators

Statistical Methods II: Cluster analysis

Objective: Determine clusters of documents that are similar to each other based on their feature vectors

- Nearest neighbor methods—K-Means, KNN
- Contextual Clustering
- Decision trees

Result: Documents can be clustered in groups that have credible substantive interpretations

Statistical Methods III: Classification algorithms

Objective: identify the characteristics of documents that are most useful in differentiating them into categories that have been specified a priori

- Discriminant analysis
- SVM
- Neural networks
- Text-specific methods such as naive Bayes, tf-idf

Result: documents can be used to classify cases into a set of categories

tf/idf: the concept

tf–idf, term frequency–inverse document frequency, is a numerical statistic which reflects how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others.

Variations of the tf–idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. tf–idf can be successfully used for stop-words filtering in various subject fields including text summarization and classification.[1]

One of the simplest ranking functions is computed by summing the tf–idf for each query term; many more sophisticated ranking functions are variants of this simple model.

Contents [show]

Motivation [edit]

Suppose we have a set of English text documents and wish to determine which document is most relevant to the query "the brown cow". A simple way to start out is by eliminating documents that do not contain all three words "the", "brown", and "cow", but this still leaves many documents. To further distinguish them, we might count the number of times each term occurs in each document and sum them all together; the number of times a term occurs in a document is called its *term frequency*.

However, because the term "the" is so common, this will tend to incorrectly emphasize documents which happen to use the word "the" more frequently, without giving enough weight to the more meaningful terms "brown" and "cow". The term "the" is not a good keyword to distinguish relevant and non-relevant documents and terms, unlike the less common words "brown" and "cow". Hence an *inverse document frequency* factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely.

Source: http://en.wikipedia.org/wiki/Tf-idf

tf/idf: the math

Mathematical details [edit]

tf–idf is the product of two statistics, term frequency and inverse document frequency. Various ways for determining the exact values of both statistics exist. In the case of the **term frequency** tf(*t*,*d*), the simplest choice is to use the *raw frequency* of a term in a document, i.e. the number of times that term *t* occurs in document *d*. If we denote the raw frequency of *t* by f(*t*,*d*), then the simple tf scheme is tf(*t*,*d*) = f(*t*,*d*). Other possibilities include[2]

- boolean "frequencies": tf(*t*,*d*) = 1 if *t* occurs in *d* and 0 otherwise;
- logarithmically scaled frequency: tf(*t*,*d*) = log (f(*t*,*d*) + 1);
- augmented frequency, to prevent a bias towards longer documents, e.g. raw frequency divided by the maximum raw frequency of any term in the document:

$$\mathrm{tf}(t,d) = 0.5 + \frac{0.5 \times \mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

The **inverse document frequency** is a measure of whether the term is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

with

- $|D|$: cardinality of D, or the total number of documents in the corpus
- $|\{d \in D : t \in d\}|$: number of documents where the term t appears (i.e., $\mathrm{tf}(t,d) \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the formula to $1 + |\{d \in D : t \in d\}|$.

Mathematically the base of the log function does not matter and constitutes a constant multiplicative factor towards the overall result.

Then tf–idf is calculated as

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D)$$

A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. Since the ratio inside the idf's log function is always greater than or equal to 1, the value of idf (and tf-idf) is greater than or equal to 0. As a term appears in more documents, the ratio inside the logarithm approaches 1, bringing the idf and tf-idf closer to 0.

Various (mathematical) forms of the tf–idf term weight can be derived from a probabilistic retrieval model that mimicks human relevance decision making.

Source: http://en.wikipedia.org/wiki/Tf-idf

Naive Bayes

Introduction [edit]

Bayesian spam filtering ·
Binary classification · **Naive Bayes classifier**

Statistics portal

V · T · E

In simple terms, a naive Bayes classifier assumes that the presence or absence of a particular feature is unrelated to the presence or absence of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 3" in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of the presence or absence of the other features.

For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods.

Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. In 2004, an analysis of the Bayesian classification problem showed that there are sound theoretical reasons for the apparently implausible efficacy of naive Bayes classifiers.[1] Still, a comprehensive comparison with other classification algorithms in 2006 showed that Bayes classification is outperformed by other approaches, such as boosted trees or random forests.[2]

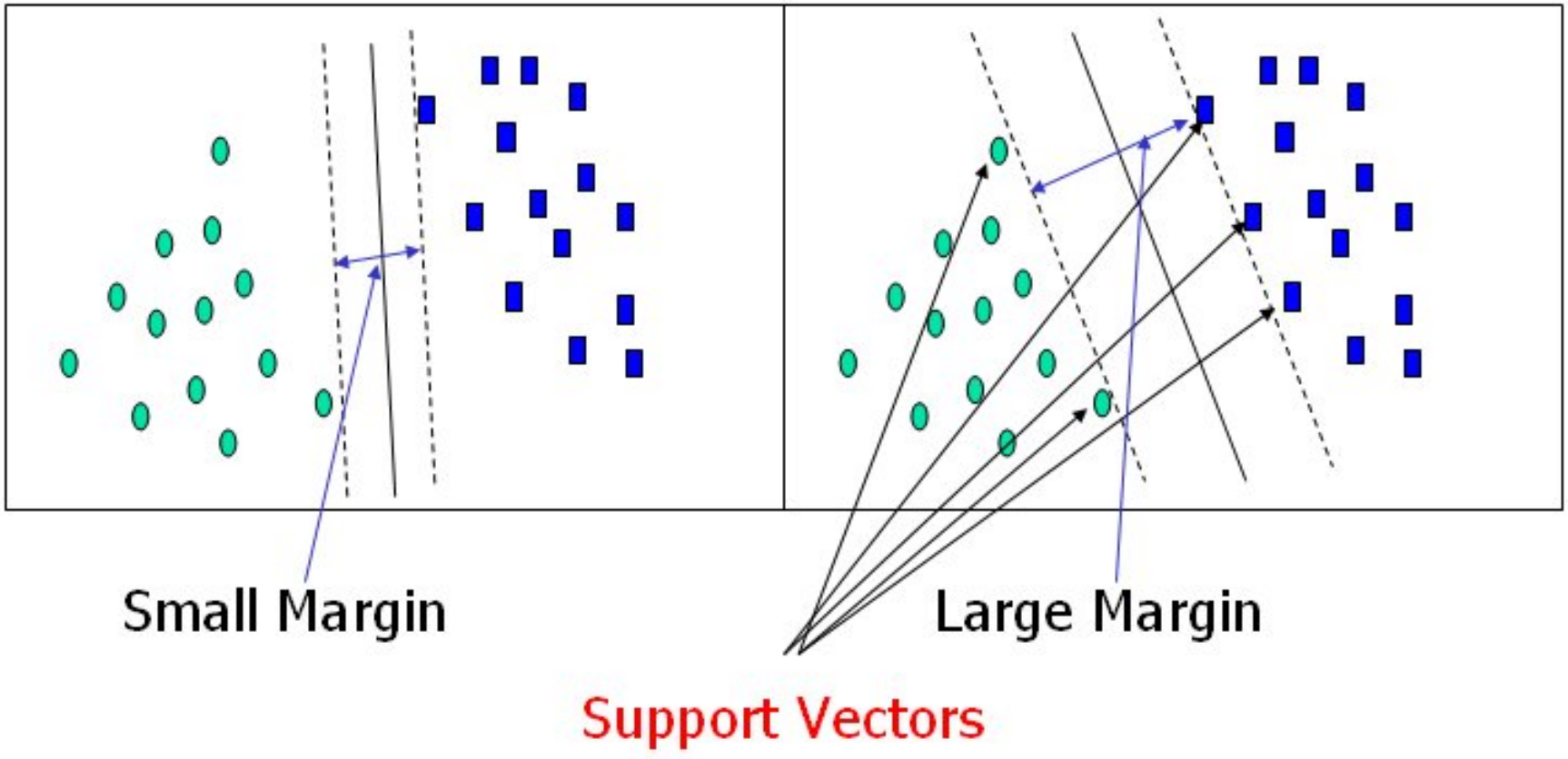
An advantage of Naive Bayes is that it only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

Source: https://en.wikipedia.org/wiki/Naive_Bayes_classifier

Support Vector Machines

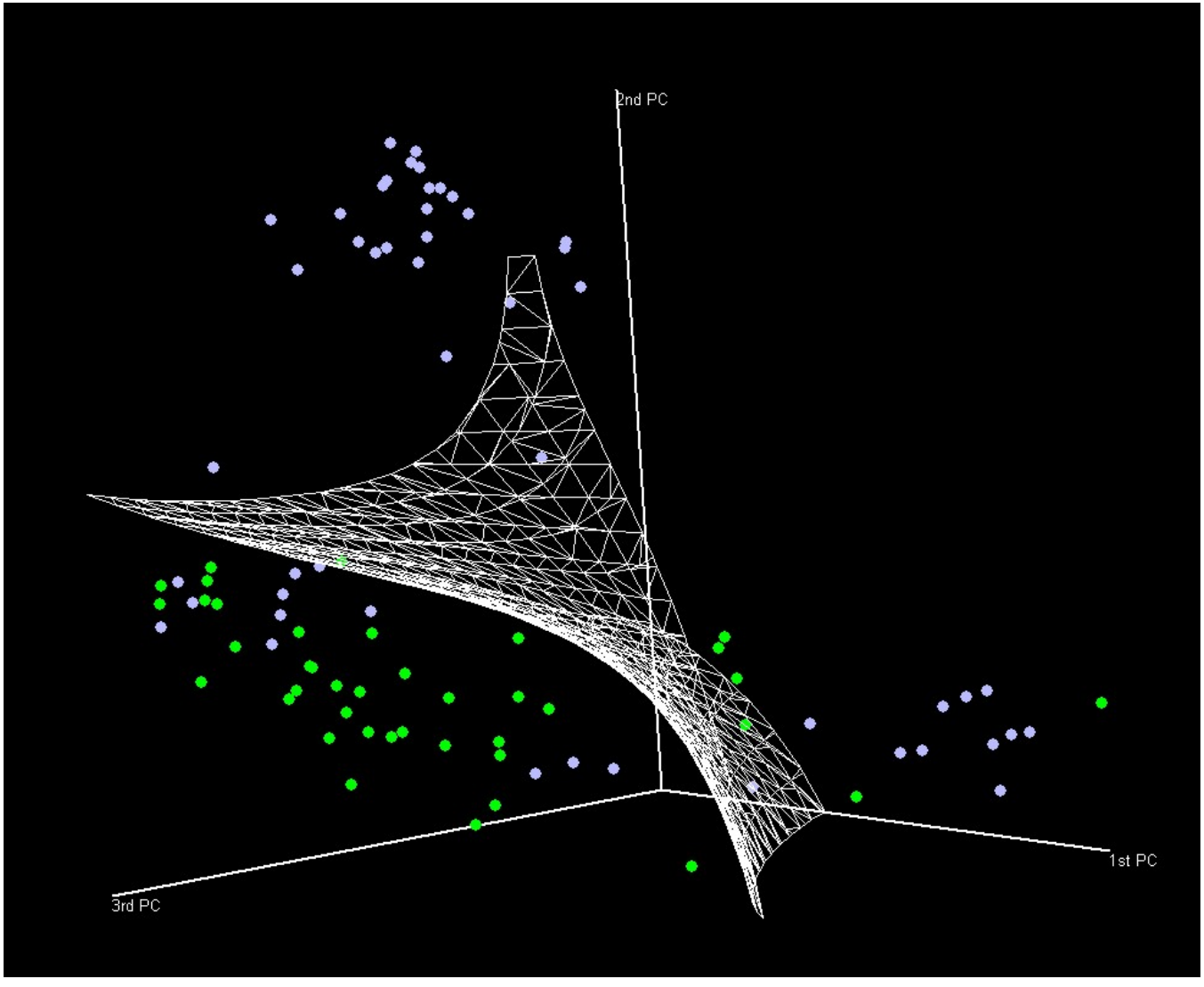
- The support-vector machine (SVM) is the workhorse of document classification and has proven to be highly robust across a wide variety of domains.
- SVM partitions a high-dimensional space into two categories while maximizing the distance between the cases at the boundary
- SVM reduces the number of “close call” cases compared to older reduction-of-dimensionality approaches such as principle components and discriminant analysis
- Multi-category SVM is done by simply setting up a series of dichotomous SVMs
- Open-source code is available in a variety of formats, including C, Java, R and MatLab

Basic SVM



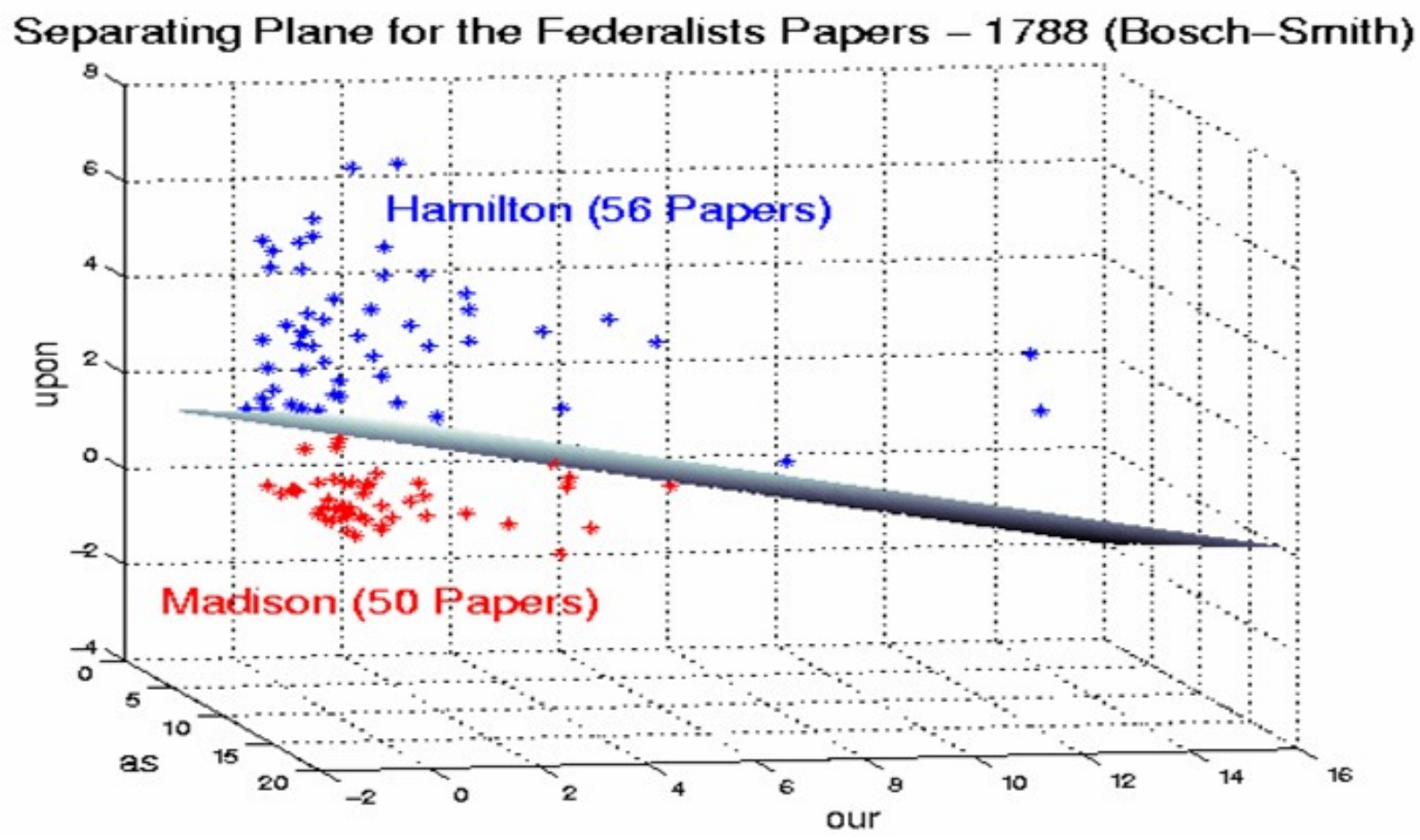
Source: http://www.dtreg.com/svm.htm

A fancier SVM



Source: http://www.epicentersoftware.com/genetrix/features/machine_learning_heuristics.htm

Applied SVM



Source: http://www.dtreg.com/svm.htm

Just one correspondence analysis graphic

since I think the method is cool...

Source:http://info.paiwhq.com/correspondence-analysis-what-does-it-all-mean/

Supervised cluster: your turn...

- What are examples of clustered behaviors in your field?
- Are there “natural” lower dimensions, or at least those used in common discussion? (note that these are not necessary, and sometimes they are illusions)
- To what extent do clusters of variables correspond to clusters of cases?

Topics: Module 3

- Distinctions between statistical and machine learning approaches
- Metrics in machine learning
- Text as a statistical object
  - Support Vector Machines
- Topic modeling: Latent Dirichlet Allocation
- Sequence models
  - Levenshtein distance
  - HMM
  - CRF
  - Additional comments on sequences

Latent Dirichlet Allocation

- Three-level hierarchical Bayesian model
- Each document is a mixture of multiple overlapping latent topics
- Each latent topic is associated with its own set of words
- Determines latent topics from documents' word-distributions
- Determines the 'composition' of a document according to topics

Latent Dirichlet Models

- Blei, Ng and Jordan (2003)
- Originally developed as a text classification method
- Assumption: texts are a composite of multiple vectors of latent topics
- Topics can be used as a data reduction method which in turn can be used as input to a classification method such as logistic regression or SVM

Hypothetical Latent Topics in Republican Candidate Speeches

Obama

Democrats

re-election

health care

socialism

Hypothetical Latent Topics in Republican Candidate Speeches

Obama	recession
Democrats	jobs
re-election	unemployment
health care	taxes
socialism	growth

Hypothetical Latent Topics in Republican Candidate Speeches

	Obama	recession
	Democrats	jobs
	re-election	unemployment
	health care	taxes
	socialism	growth
Rom	50%	50%
Paw	40%	60%

Hypothetical Latent Topics in Republican Candidate Speeches

	Obama	recession	Alaska
	Democrats	jobs	moose
	re-election	unemployment	Russia
	health care	taxes	guns
	socialism	growth	ya'know?
Rom	50%	50%	
Paw	40%	60%	
Pal	30%	35%	35%

Hypothetical Latent Topics in Republican Candidate Speeches

	Obama	recession	Alaska	I
	Democrats	jobs	moose	me
	re-election	unemployment	Russia	myself
	health care	taxes	guns	Trump
	socialism	growth	ya'know?	
Rom	50%	50%		
Paw	40%	60%		
Pal	30%	35%	35%	
Tru	10%	10%		80%

LDA on (Politically Relevant) News-Story Texts

Benefits...

- Stories can deal with multiple, simultaneous topics
- Flexible categories and classifications
- Approach is inductive and could identify under-represented categories of actions
- Allows for general classification of stories when verb-phrase dictionaries don't exist
- Language neutral: requires only word frequency and does not need syntactical rules

Applying LDA to ICEWS News-Story Texts

Some questions we hope to answer...

- Are latent topics similar across countries?
- Are latent topics similar to those used by in event data coding schemes (WEIS, CAMEO, IDEA)?
- Are there topics which are common in the news stories but not coded in the event data schemes?
- How many non-political topics (sports; arts and entertainment) do we find?
- Do the topics cluster plausibly?

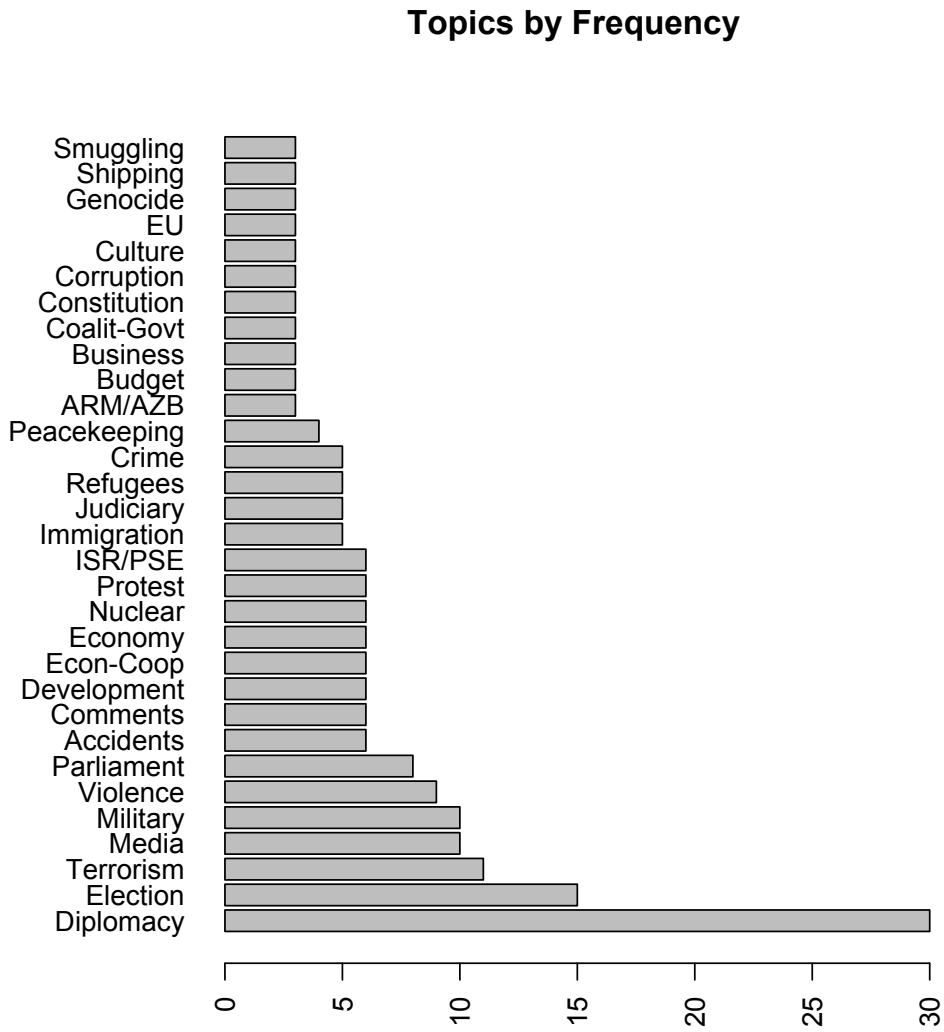
ICEWS News-Story Sample

- 61 European and Middle Eastern countries
- January 2001 to July 2011
- Document: An individual news-story
- Two “levels” of news-story corpora:
  - 61 separate country-corpora
  - 1 combined country-corpus, with random sample of news-stories

Application

- Preprocessing:
  - Stem words
  - Remove: punctuation, whitespace, stopwords, numbers
  - Remove proper nouns using CountryInfo.txt
- LDA
  - Unsupervised LDA in R
  - Set topics to 10 and then to 20
  - Store topics-matrices for each corpus
- Detailed analysis
  - Combined sample (“ALL”)
  - Europe: France, Greece, Norway, Poland
  - Middle East: Egypt, Israel, Jordan, Turkey
  - Other: Albania, Armenia, Portugal

Topics by Frequency



Top Words for All-Countries (Topics 1-10 of 1-20)

Negotiation	Comments	Protest	Media	Accidents
talk	world	protest	report	hospit
meet	war	demonstr	newspap	kill
peac	global	violenc	conflict	plane
summit	newspap	peopl	tv	crash
negoti	women	opposit	resolut	peopl
visit	via	attack	interview	die
leader	look	condemn	territori	injur
discuss	write	ralli	publish	polic

Military	Econ-Coop	Diplomacy	Nuclear	Democracy
militari	cooper	agenc	nuclear	polit
defenc	visit	news	weapon	reform
forc	meet	report	sanction	polici
troop	trade	sourc	energi	countri
defens	develop	meet	secur	democrat
command	econom	visit	resolut	democraci
arm	discuss	excerpt	power	chang
oper	agreement	web	atom	futur

Top Words for France (Topics 1-10 of 1-20)

Parliament	Diplomacy	Violence	Crime	Terrorism
elect	cooper	kill	court	attack
parti	relat	attack	crime	terrorist
democrat	meet	forc	charg	arrest
vote	countri	soldier	prison	suspect
opposit	develop	troop	trial	terror
candid	bilater	rebel	war	polic
parliamentari	region	fire	sentenc	bomb
coalit	visit	bomb	prosecutor	kill

Election	Parliament	Ceremony	Economy	Smuggling
vote	law	ceremoni	bank	border
percent	right	anniversari	percent	polic
elect	agreement	visit	fund	drug
parti	visa	cultur	budget	ship
referendum	amend	church	compani	illeg
parliament	human	celebr	oil	guard
poll	refuge	peopl	tax	cross
govern	draft	attend	market	traffick

General Topics from Combined Sample

Negotiation	talk meet peac summit negoti visit leader discuss minist middl issu diplomat offici presid agreement foreign agre process plan confer
Military	militari defenc forc troop defens command arm oper air mission peacekeep deploy soldier armi train secur exercis base missil staff
Violence	kill attack forc soldier troop rebel fire bomb wound ethnic milit villag militari border armi provinc citi civilian polic northern
Parliament	elect parti democrat vote opposit candid parliamentari coalit presidenti parliament poll deputi polit chairman leader presid seat voter socialist rule
Election	vote percent elect parti referendum parliament poll govern constitut coalit opposit join reform bloc treati minist voter polit candid democrat
Economy	bank percent fund budget compani oil tax market economi financ financi price gas econom product rate export money growth invest
Crime	court crime charg prison trial war sentenc prosecutor former investig tribun arrest alleg accus judg lawyer murder crimin suspect convict

Country-Specific Topics

Culture	FRA	film ceremoni wife cultur celebr book world award life art histori anniversari festiv presid famili memori centuri love honour former
Nobel-Prize	NOR	peac award prize right human committe ceremoni winner world nomin dissid laureat former presid won campaign win democraci ja
Royals	NOR	children royal school celebr coupl famili visit citi student hospit princess live ceremoni sonja mother home wife church life father
Gaza	ISR	border ship forc troop aid flotilla plane blockad cross raid southern deploy armi south smuggl militari weapon peacekeep air activist
Cyprus	TUR	island northern solut denkta republ plan talk leader negoti divid settlement issu peac reunif communiti north meet agreement referer
Cyprus	GRC	island talk leader plan divid denkta solut northern negoti republ pea reunif settlement north meet coup communiti referendum reunit

Topics: Module 3

- Distinctions between statistical and machine learning approaches
- Metrics in machine learning
- Text as a statistical object
  - Support Vector Machines
- Topic modeling: Latent Dirichlet Allocation
- Sequence models
  - Levenshtein distance
  - HMM
  - CRF
  - Additional comments on sequences

General approach to sequence modeling

- Sequence is defined by a finite set of possible symbols
- Series of operations or rules for going between the symbols
- Applications
  - Spell checking
  - Parts of speech tagging
  - Spoken language recognition
  - Genomics: DNA and amino acid sequences
  - Careers of political activists
  - Transitions between authoritarianism and democracy

Levenshtein distance

- Distance between two strings/sequences is the operations which combine to the minimum cost
  - Insertion: vector of costs by symbol
  - Deletion: vector of costs by symbol
  - Substitution: matrix of costs by symbol x symbol
- This is computed using a relatively efficient dynamic programming algorithm
- CRAN: ‘lwr’, ‘stringdist’
- http://en.wikipedia.org/wiki/Levenshtein_distance

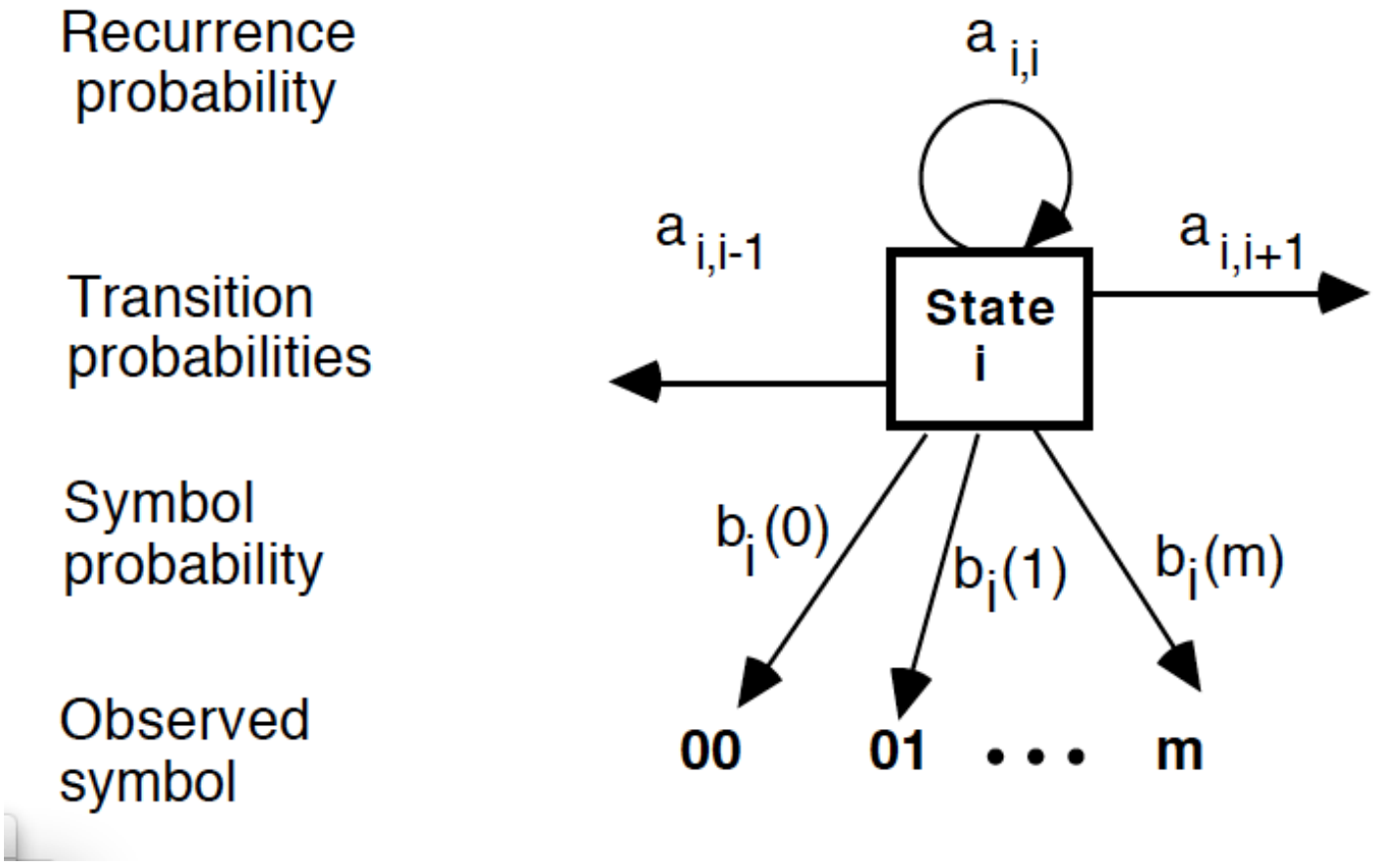
Levenshtein distance between “kitten” and “sitting”

1. kitten $\rightarrow$ sitten (substitution of ‘s’ for ‘k’)
2. sitten $\rightarrow$ sittin (substitution of ‘i’ for ‘e’)
3. sittin $\rightarrow$ sitting (insertion of ‘g’ at the end).

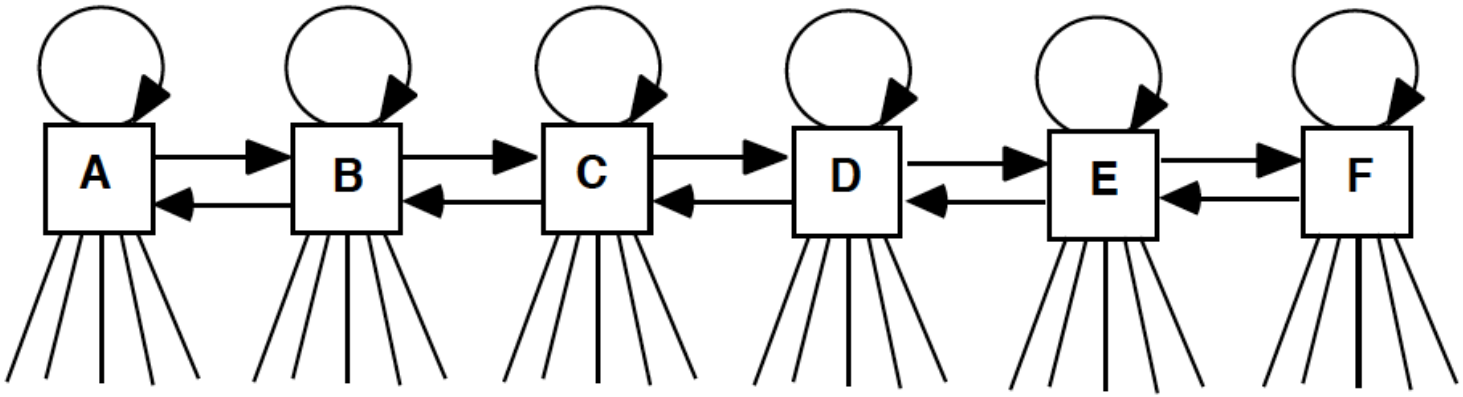
Hidden Markov Model

- Markov assumption: transition between states of the system are a function of only the current state and the transition matrix
- Application: crisis phase
- States are not directly observed—hence "hidden"—but each state is associated with a probability distribution of the symbols generated by the system
- The transition matrix and probabilities are estimated using the Baum-Welch expectation-maximization algorithm. There are multiple packages on CRAN for this. Major problem is local maxima in this estimation.
- Training is by example
- The Viterbi algorithm can be used to establish the likely sequence of states given an observed set of symbols
- Typical application is to match an observed set of symbols to a series of models and then choose the models which had the maximum probability
- These probabilities are proportional to the length of the sequence, so it is difficult to compare fits sequences of different

An element of a left-right-left hidden Markov model



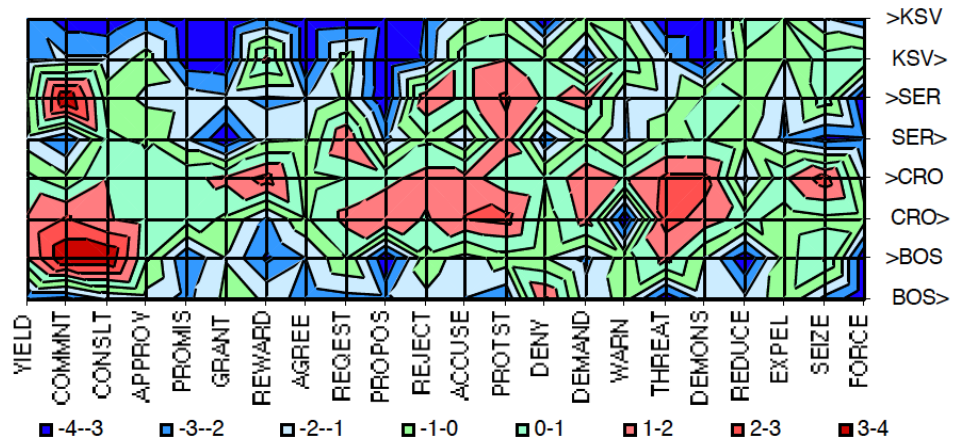
A left-right-left (LRL) hidden Markov Model



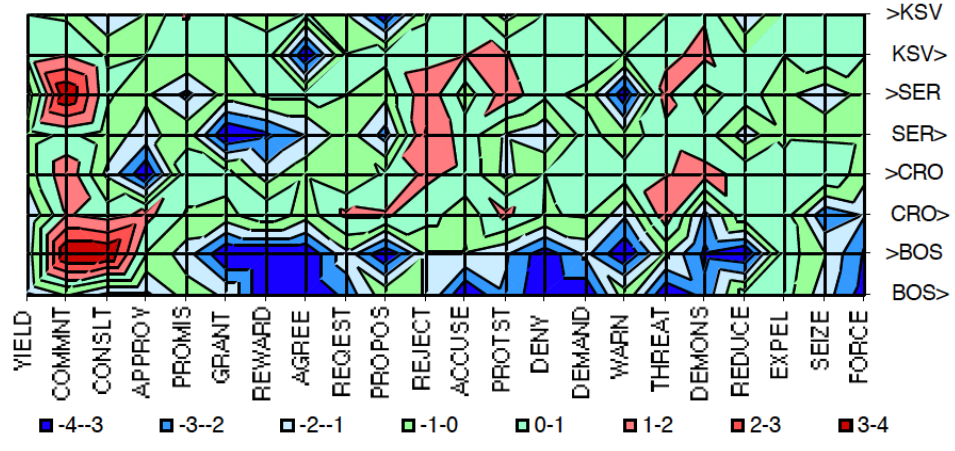
HMM probability map for Balkans

Figure 13b
DIFFERENCE-OF-MEANS TESTS BETWEEN ESTIMATED AND MARGINAL PROBABILITIES, 3-MONTH LOW MODELS STATE 1

P3 - LOW - STATE 1



N3 - LOW - STATE 1



Conditional Random Fields

- In a CRF, each feature function is a function that takes in as input:
  - a sentence s
  - the position i of a word in the sentence
  - the label l_i of the current word
  - the label l_{i-1} of the previous word
- Each of these items is associated with a weight, which is estimated. Information from additional locations in the sequence can also be used.
- The CFR outputs a real-valued number (though the numbers are often just either 0 or 1
- CRFs are basically the sequential version of logistic regression: whereas logistic regression is a log-linear model for classification, CRFs are a log-linear model for sequential labels.

Source: http://blog.echen.me/2012/01/03/introduction-to-conditional-random-fields/

Conditional Random Fields

This is more general than an HMM:

- CRFs can define a much larger set of features. HMMs are necessarily local in nature, which force each word to depend only on the current label and each label to depend only on the previous label. CRFs can use more global features.
- CRFs can have arbitrary weights. Whereas an HMM uses probabilities

Complications

- Sequences may not have a strict ordering when multiple preconditions are running in parallel and can be completed in any order
- Sequences tend to occur in ordinal rather than interval time: are "non-events" important?
- The computational time for these methods tends to be proportional to the sequence of the sequence length

Sequences: your turn...

- What are examples of sequenced behaviors in your field?
- What would the symbol sets look like?
- What might determine the weights?

Let's eat!

End of Module 3