Predictive Analytics
Machine Learning and Sequence-based Methods

Philip A. Schrodt

Parus Analytical Systems
schrodt735@gmail.com

Odum Institute "Data Matters" Workshop
University of North Carolina
26 June 2014

Inspiration

UCBerkeleyNews

NewsCenter

▶ Today's news & events

▶ Subscribe to news

▶ For the news media

▶ Calendar of events

Top stories:

Cal Dining team goes for glory at Gilroy Garlic Festival

Three young researchers named 2014 Pew Scholars

Berkeley physicists detect smallest force ever measured

Young researcher discovers source of disco clams' light show

Researcher calls report on economic impacts of U.S. climate change 'like a flashlight at night'

Browse by Subject or Date Select one News Search Google™ Custom Se Go

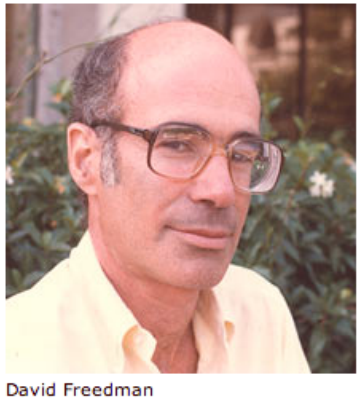
Press Release

E-mail story

Noted statistician David Freedman has died at 70

By Robert Sanders, Media Relations | 20 October 2008

BERKELEY — David A. Freedman, a professor of statistics at the University of California, Berkeley, who fought for three decades to keep the United States census on a firm statistical foundation, died Friday, Oct. 17, of bone cancer at his home in Berkeley. He was 70.



David Freedman

Throughout his career, Freedman made major contributions to the theory and teaching of statistics. But he also had a broad impact on the application of statistics to important medical, social, legal and public policy issues, including clinical drug trials, epidemiologic studies, economic models, interpretation of scientific experiments, statistical evidence in the courtroom and adjustments to the census.

"David transformed the practice of applied statistics as it is directed toward litigation, toward Congressional action and toward public policy," said long-time friend and colleague Kenneth Wachter, UC Berkeley professor of demography and statistics. "The prevailing mode when he began working was to rely on hypothetical models with assumptions sometimes driven by mathematical convenience, which were fine for theoretic work but, when carried over to applications in the policy arena, gave conclusions that were often fanciful or driven by the prejudices or presuppositions of the statisticians testifying or contributing."

"Not only has David, since his early twenties, been recognized as one of the world's leading mathematical statisticians, but he has also assumed the mantle as the skeptical conscience of statistics as it is applied to important scientific, policy and legal issues," wrote James M. Robins, professor of epidemiology at the Harvard School of Public Health, in 2002.

Freedman clarified the assumptions underlying a wide variety of statistical models and revealed how sensitive conclusions can be to violations of the those assumptions - regardless of the quality of the data. "By distinguishing proposals based on hypothetical modeling from proposals grounded in empirically established observations, he developed a firmer basis for applying statistics to policy," Wachter said.

His legacy, said UC Berkeley colleague Philip Stark, professor of statistics, is "demystifying and debunking many of the tools people use in social science and elsewhere to try to draw inferences." Even today, "there is a lot of muddled thinking and blind reliance on methodology - almost a religious belief that methods give truth - without looking carefully at the assumptions of the methodology. David contributed enormously to the clarity and rigor and circumspection" in the field of applied statistics.

Major lessons learned so far

- There are strong theoretical reasons to believe that error cannot be reduced to zero, but there is no reason why it is stabilizing at 80%
- Successful models are generally relatively simple
- Multiple methods generally converge to similar levels of accuracy, though there are probably minor gains to be made by refining these methods
- Ensemble methods are proving successful for both technical and human forecasts
- Event-based and structural models are *probably* substitutable at relatively short time frames

Partial autocorrelation function

From Wikipedia, the free encyclopedia

This article includes a list of references, related reading or external links, but **its sources remain unclear because it lacks inline citations**. Please improve this article by introducing more precise citations. *(September 2011)*

In time series analysis, the **partial autocorrelation function** (**PACF**) plays an important role in data analyses aimed at identifying the extent of the lag in an autoregressive model. The use of this function was introduced as part of the Box–Jenkins approach to time series modelling, where by plotting the partial autocorrelative functions one could determine the appropriate lags **p** in an AR (**p**) model or in an extended ARIMA (**p**,**d**,**q**) model.

Description [edit]

Given a time series z_t, the partial autocorrelation of lag *k*, denoted $\alpha(k)$, is the autocorrelation between z_t and z_{t+k} with the linear dependence of z_{t+1} through to z_{t+k-1} removed; equivalently, it is the autocorrelation between z_t and z_{t+k} that is not accounted for by lags 1 to $k-1$, inclusive.

$$\alpha(1) = \operatorname{Cor}(z_t, z_{t+1})$$
$$\alpha(k) = \operatorname{Cor}(z_{t+k} - P_{t,k}(z_{t+k}),\ z_t - P_{t,k}(z_t)),\ \text{for } k \geq 2,$$

where $P_{t,k}(x)$ denotes the projection of x onto the space spanned by $z_{t+1}, \ldots, z_{t+k-1}$.

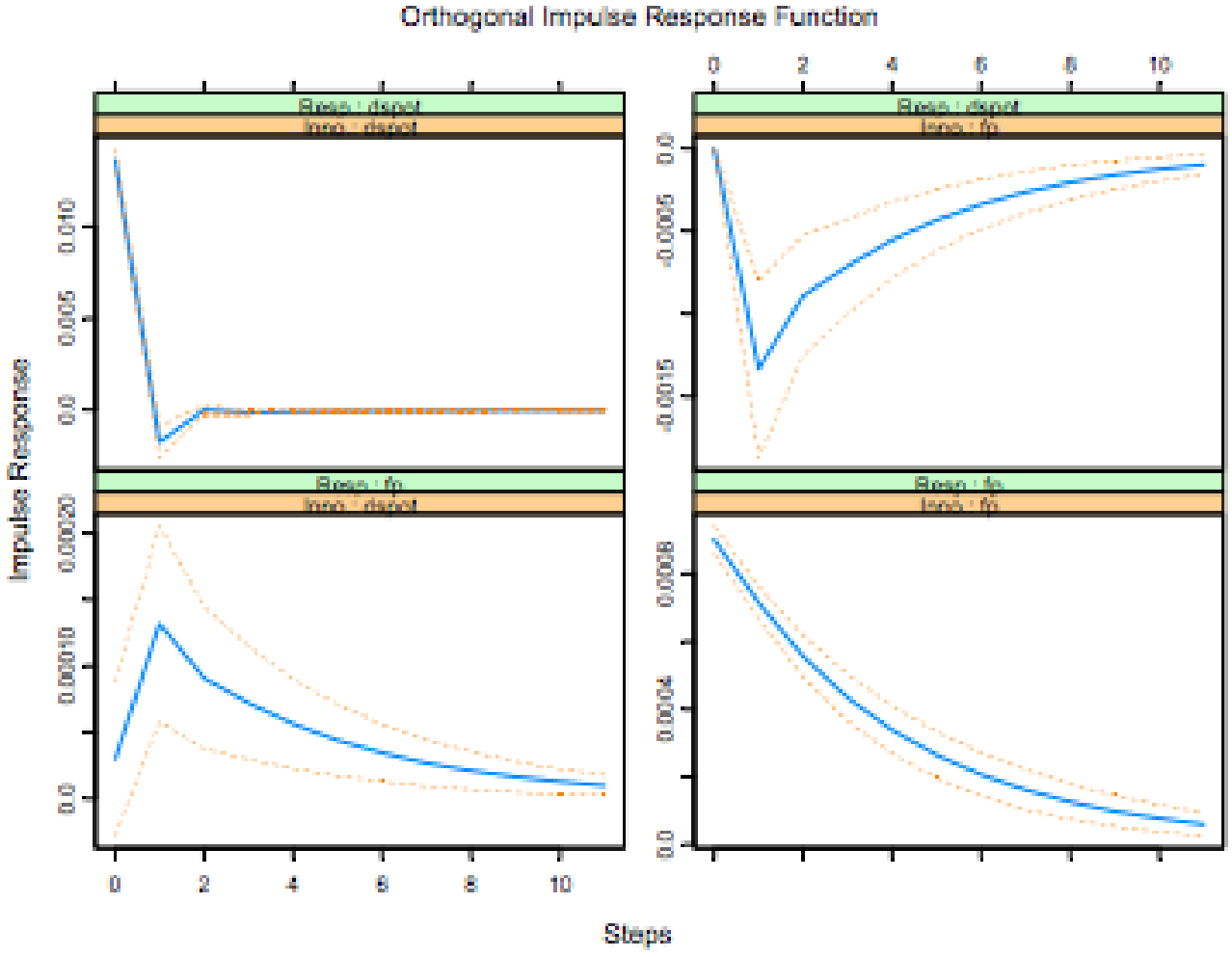
There are algorithms, not discussed here, for estimating the partial autocorrelation based on the sample autocorrelations. See (Box, Jenkins, and Reinsel 2008) or (Brockwell and Davis, 2009) for the mathematical details. These algorithms derive from the exact theoretical relation between the partial autocorrelation function and the autocorrelation function.

Granger Causality and Vector Autoregression

Y is "Granger-caused" by X when the prediction of Y by the lagged values of X and Y is better than the prediction by the lagged values of Y alone.

Vector Autoregression (VAR)
Essentially use a Granger approach, and pay no attention to the coefficient values because of the effects of autocorrelation and colinearity. Instead look at the effect of a shock to the variable. Widely used by the U.S. Federal Reserve.

VAR: Response functions



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-1

- Focus on out-of-sample validation, not standard error of coefficients
  - Out-of-sample is also needed because of the danger of overfitting
- Collinearity is an asset, not a liability
- Assumption—and exploitation—of heterogeneous subpopulations
- Missing values can be data
- Sparse datasets: most indicators are not measured on most cases

Distinctions between statistical and machine learning approaches-2

- Non-linear, and consequently the *cases>>variables* constraint need not apply
- Diffuse knowledge/coefficient structures: VAR, BMA, neural networks, random forests, and HMM/CRF
- ML methods are frequently just the application of a "common sense" algorithm, whereas statistical approaches often require detailed mathematical derivations and the properties may be dependent on unrealistic—or unknowable—properties of the data

Available software

R

Very conveniently, R has emerged as a very common tool in machine learning. Even when it doesn't necessarily make sense.

Weka Project

Machine Learning Group at the University of Waikato

Project | **Software** | Book | Publications | People | Related

Weka 3: Data Mining Software in Java

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

Found only on the islands of New Zealand, the Weka is a flightless bird with an inquisitive nature. The name is pronounced like **this**, and the bird sounds like **this**.

Weka is open source software issued under the **GNU General Public License**.

Getting started

- **Requirements**
- **Download**
- **Documentation**
- **FAQ**
- **Getting Help**

Further information

- **Citing Weka**
- **Datasets**
- **Related Projects**
- **Miscellaneous Code**
- **Other Literature**

Developers

- **Development**
- **History**
- **Subversion**
- **Contributors**

Weka features

Weka's main user interface is the *Explorer*, but essentially the same functionality can be accessed through the component-based *Knowledge Flow* interface and from the command line. There is also the *Experimenter*, which allows the systematic comparison of the predictive performance of Weka's machine learning algorithms on a collection of datasets.

The *Explorer* interface features several panels providing access to the main components of the workbench:

- The *Preprocess* panel has facilities for importing data from a database, a CSV file, etc., and for preprocessing this data using a so-called *filtering* algorithm. These filters can be used to transform the data (e.g., turning numeric attributes into discrete ones) and make it possible to delete instances and attributes according to specific criteria.
- The *Classify* panel enables the user to apply classification and regression algorithms (indiscriminately called *classifiers* in Weka) to the resulting dataset, to estimate the accuracy of the resulting predictive model, and to visualize erroneous predictions, ROC curves, etc., or the model itself (if the model is amenable to visualization like, e.g., a decision tree).
- The *Associate* panel provides access to association rule learners that attempt to identify all important interrelationships between attributes in the data.
- The *Cluster* panel gives access to the clustering techniques in Weka, e.g., the simple k-means algorithm. There is also an implementation of the expectation maximization algorithm for learning a mixture of normal distributions.
- The *Select attributes* panel provides algorithms for identifying the most predictive attributes in a dataset.
- The *Visualize* panel shows a scatter plot matrix, where individual scatter plots can be selected and enlarged, and analyzed further using various selection operators.

Source: http://en.wikipedia.org/wiki/Weka_(machine_learning)

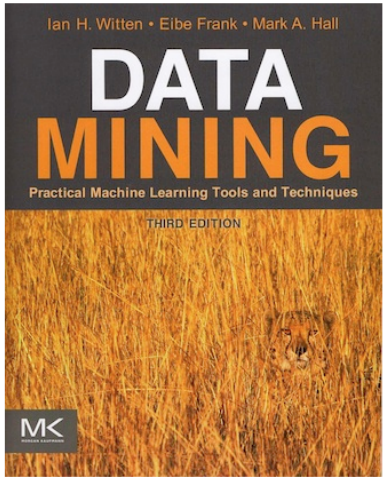
Weka Project: Data Mining

Machine Learning Group at the University of Waikato

Project Software **Book** Publications People Related

Data Mining:
Practical Machine Learning Tools and Techniques

We have written a companion book for the Weka software, now into its third edition, that describes the machine learning techniques that it implements and how to use them. It is structured into three parts. The first part is an introduction to data mining using basic machine learning techniques, the second part describes more advanced machine learning methods, and the third part is a user guide for Weka. The third edition was published in January 2011 by Morgan Kaufmann Publishers (ISBN: 978-0-12-374856-0). **Mark Hall** has joined **Ian Witten** and **Eibe Frank** as co-author for this edition, which has expanded to 629 pages.



Click here to order from Amazon.com

"If you have data that you want to analyze and understand, this book and the associated Weka toolkit are an excellent way to start."

-Jim Gray, Microsoft Research

"The authors provide enough theory to enable practical application, and it is this practical focus that separates this book from most, if not all, other books on this subject."

-Dorian Pyle, Director of Modeling at Numetrics

"This book would be a strong contender for a technical data mining course. It is one of the best of its kind."

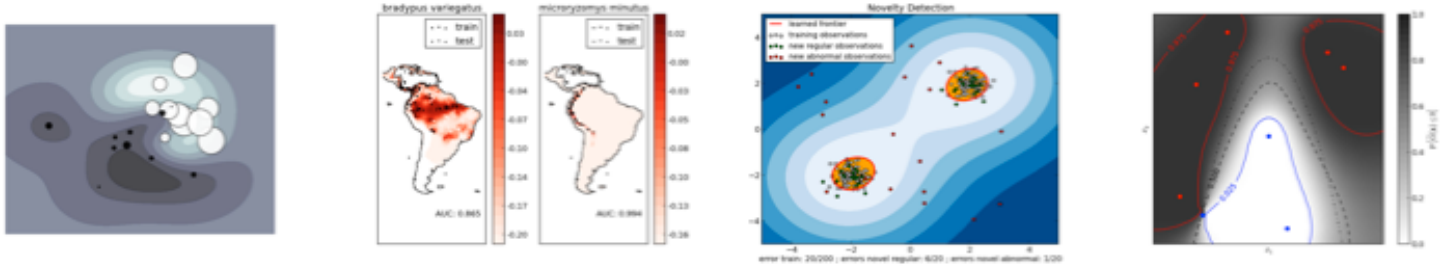
-Herb Edelstein, Principal, Data Mining Consultant, Two Crows Consulting

"It is certainly one of my favourite data mining books in my library."

-Tom Breur, Principal, XLNT Consulting, Tiburg, Netherlands

Python: scikit-learn

scikit-learn: machine learning in Python



Easy-to-use and general-purpose machine learning in Python

Scikit-learn integrates **machine learning** algorithms in the tightly-knit scientific **Python** world, building upon numpy, scipy, and matplotlib. As a machine-learning module, it provides versatile tools for data mining and analysis in any field of science and engineering. It strives to be **simple and efficient**, accessible to everybody, and reusable in various contexts.

Supervised learning

Support vector machines, linear models, naive Bayes, Gaussian processes...

Unsupervised learning

Clustering, Gaussian mixture models, manifold learning, matrix factorization, covariance...

And much more

Model selection, datasets, feature extraction... ***See below.***

License: Open source, commercially usable: **BSD license** (3 clause)

Source: http://scikit-learn.org/stable/

Metrics

Time series predictions and probabilities

- Levels of a continuous variable: classical time series methods
- Point predictions within a given time interval: logistic
  - This is the single most common approach, but a variety of different methods are being used
  - Poisson and negative binomial regression might be relevant here but high autocorrelation violates of the assumption of independence
- Point-prediction with a distribution
- Response of system to external shocks: vector autoregression
- Likelihood of an event as a function of time: Survival/hazard models
- Phase models: Bayesian switching models, hidden Markov, conditional random fields

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

$$Accuracy = \frac{number\ of\ correct\ predictions}{total\ predictions\ made}$$

$$Recall = \frac{number\ of\ correctly\ predicted\ conflicts}{total\ conflicts\ that\ occured}$$

$$Precision = \frac{number\ of\ correctly\ predicted\ conflicts}{total\ conflicts\ predicted}$$

"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)

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

Additional classification matrix-based measures

True negative rate $= \frac{tn}{tn+fp}$ (also called "Specificity")

Ratio of true positives to false positives $= \frac{tp}{fp}$

F1 score

The traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall:
$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$.
The general formula for positive real β is:
$F_\beta = (1 + \beta^2) \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{(\beta^2 \cdot \mathrm{precision}) + \mathrm{recall}}$.
The formula in terms of Type I and type II errors:
$F_\beta = \frac{(1+\beta^2) \cdot \mathrm{true\ positive}}{((1+\beta^2) \cdot \mathrm{true\ positive} + \beta^2 \cdot \mathrm{false\ negative} + \mathrm{false\ positive})}$
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

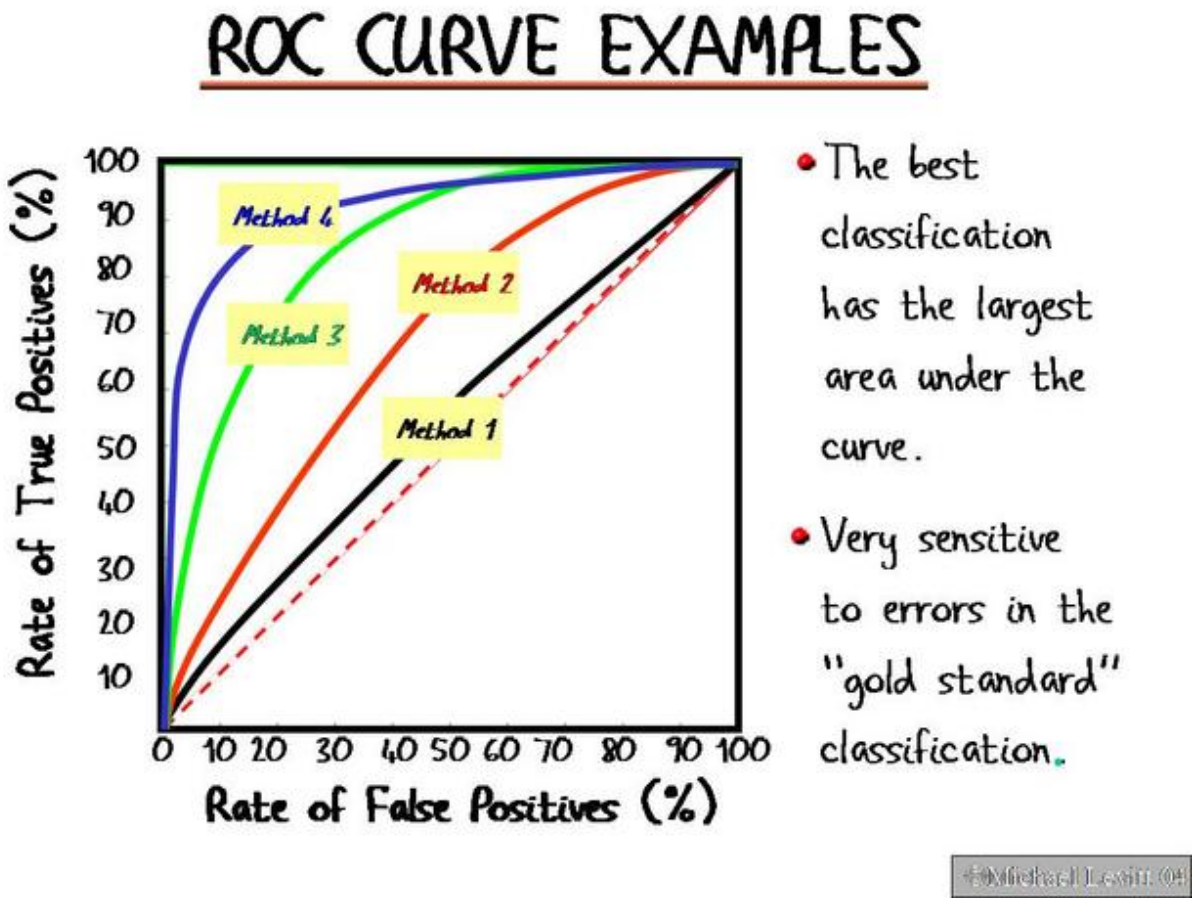
		true		
pred	0	1	row N	
0	904	592	1496	
1	126	283	409	
col N	1030	875	1905	
	Acc	0.623	AUC	0.732
	Spec	0.604	Sens	0.691
	Prec	0.323	F1	0.421
sLDA AUC		0.527		
ICEWS Reference Model				
	Acc	0.852		
	Spec	0.996		
	Sens	0.387		
	N	4437		

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

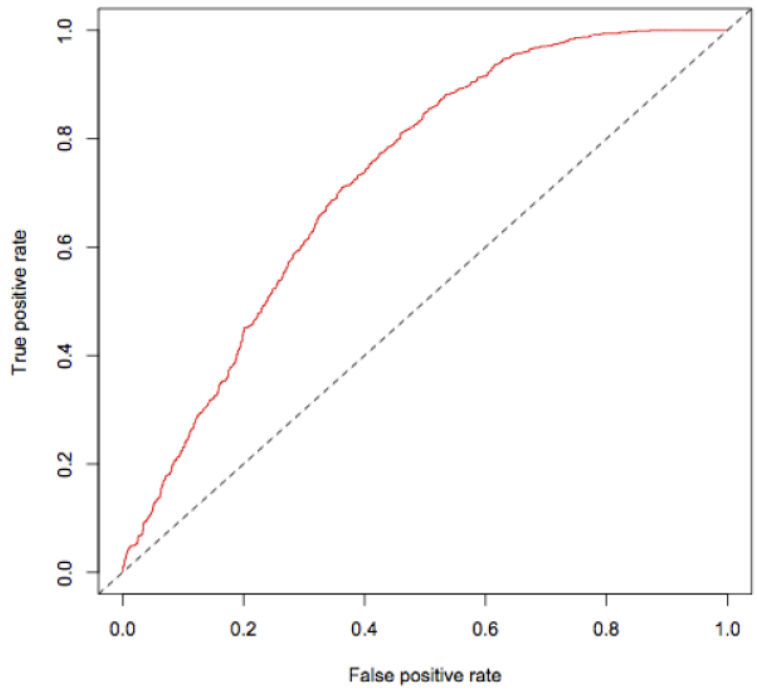
ROC Curve



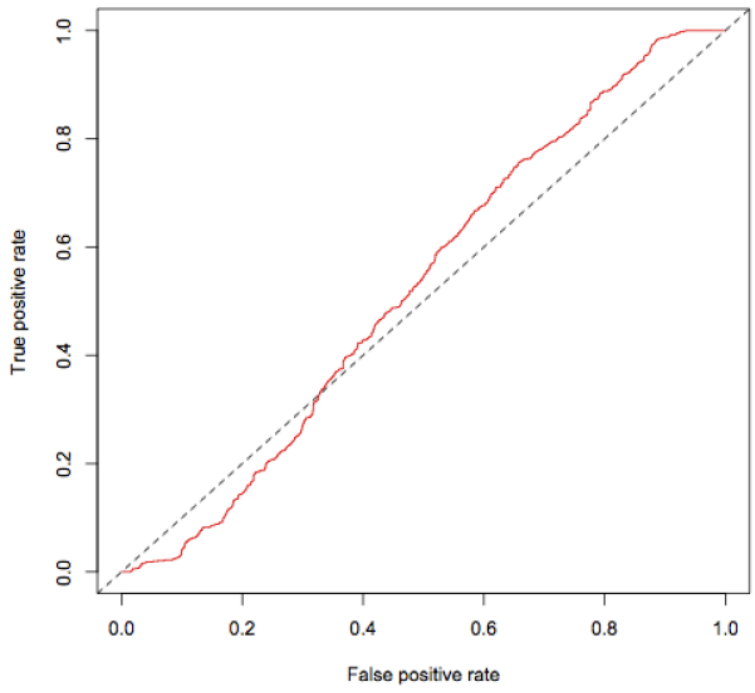
Source:
http://csb.stanford.edu/class/public/lectures/lec4/Lecture6/Data_Visualization/images/Roc_Curve_Exa

ROC Curve

LDA ROC: REBELL



ROC Curve


sLDA ROC: REBELL
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

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

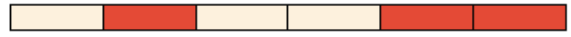
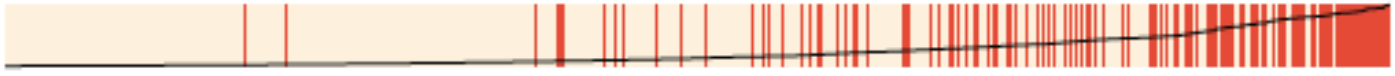


FIGURE 3 Separation Plot for a Larger Data Set

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

Machine Learning I: Cluster analysis

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

- Discriminant analysis
- Nearest neighbor methods—K-Means, KNN
- Support vector machines

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

Machine Learning II: Classification algorithms

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

- Decision trees: ID3, C4.5
- Random forests^{TM}
- Neural networks
  - These did not work in the State Failures Project but in general are a useful "Big Data" tool, so it is [very] possible that they were simply implemented badly

Result: Cases can be used to classify cases into a pre-determined set of categories

Machine Learning III: Sequence algorithms

Objective: identify the characteristics of cases based on the sequence of events. This attempts to mimic the "episodic memory/recognition" that appears to be hard-wired in humans, but also is similar to methods used in biological and linguistic pattern recognition

- rule-based models
- Levenshtein metric
- hidden Markov models and conditional random fields
- biological sequence recognition

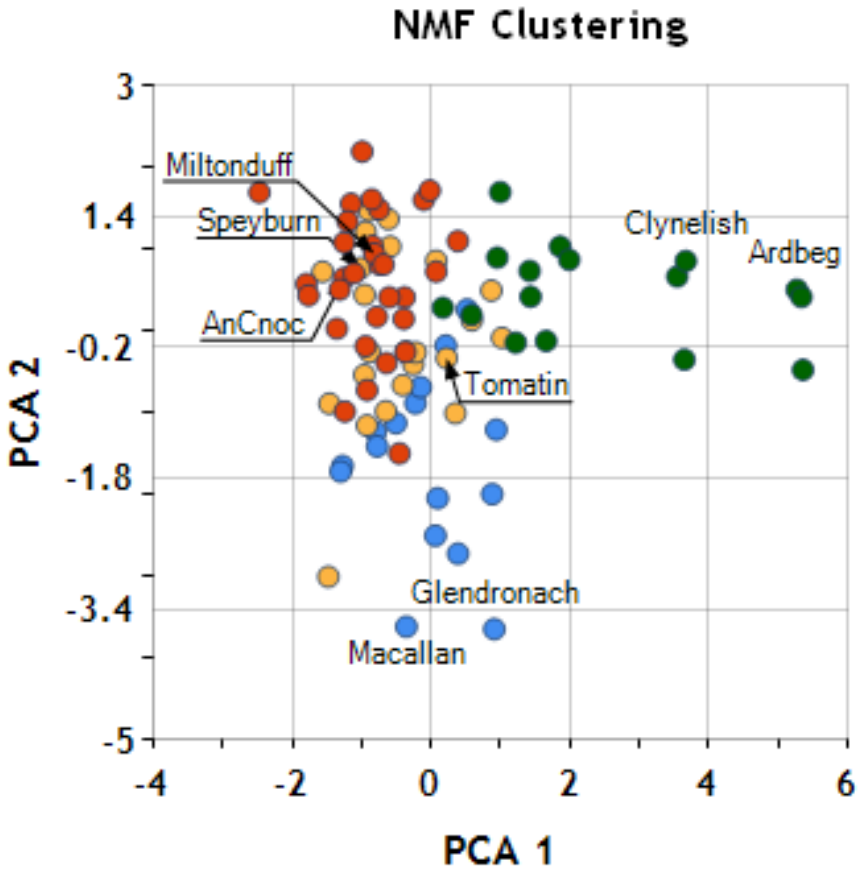
Result: Case can be compared explicitly as sequences and those comparisons can be used, typically as distance metrics, in other methods

Clustering approaches

General comments

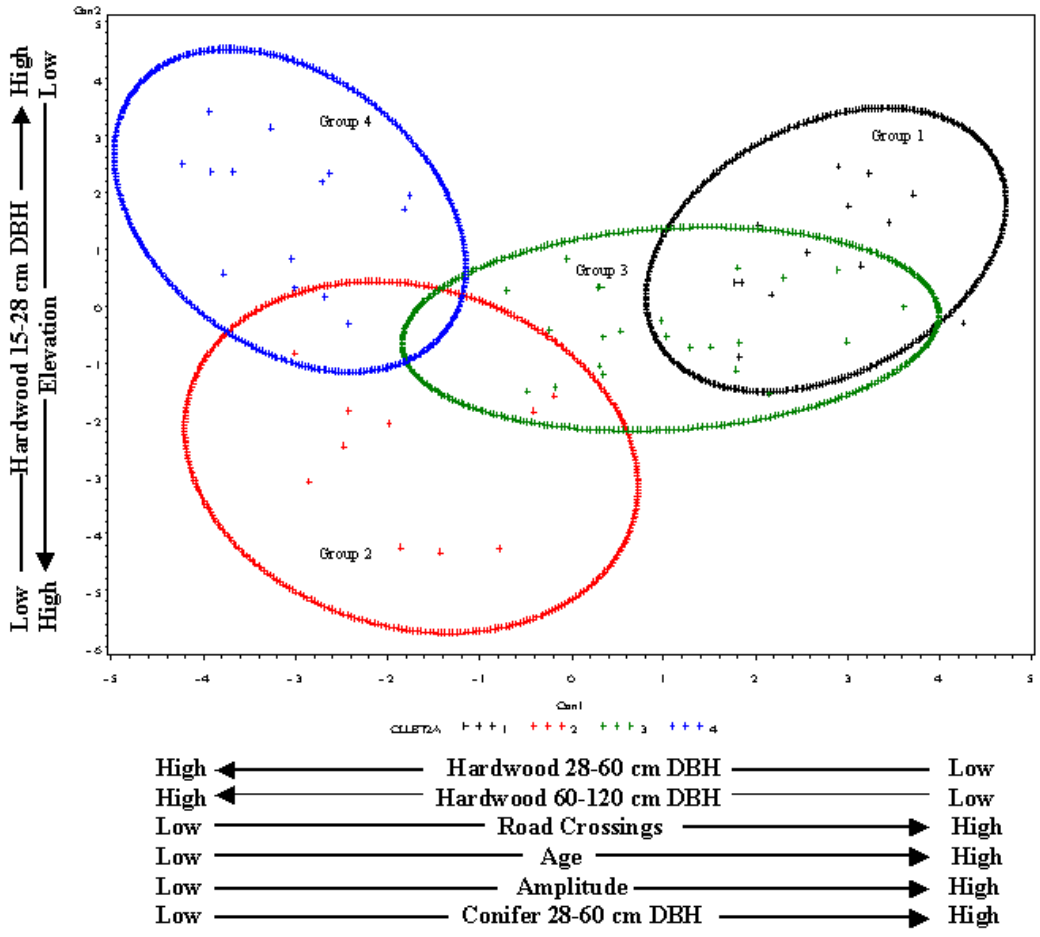
- Requires a metric—and there are many—for the distance between the cases
- In contrast to linear approaches this *assumes* heterogeneous subpopulations
- Clustering is typically depicted in two dimensions but usually is computed in an arbitrarily large space

Cluster Example 1



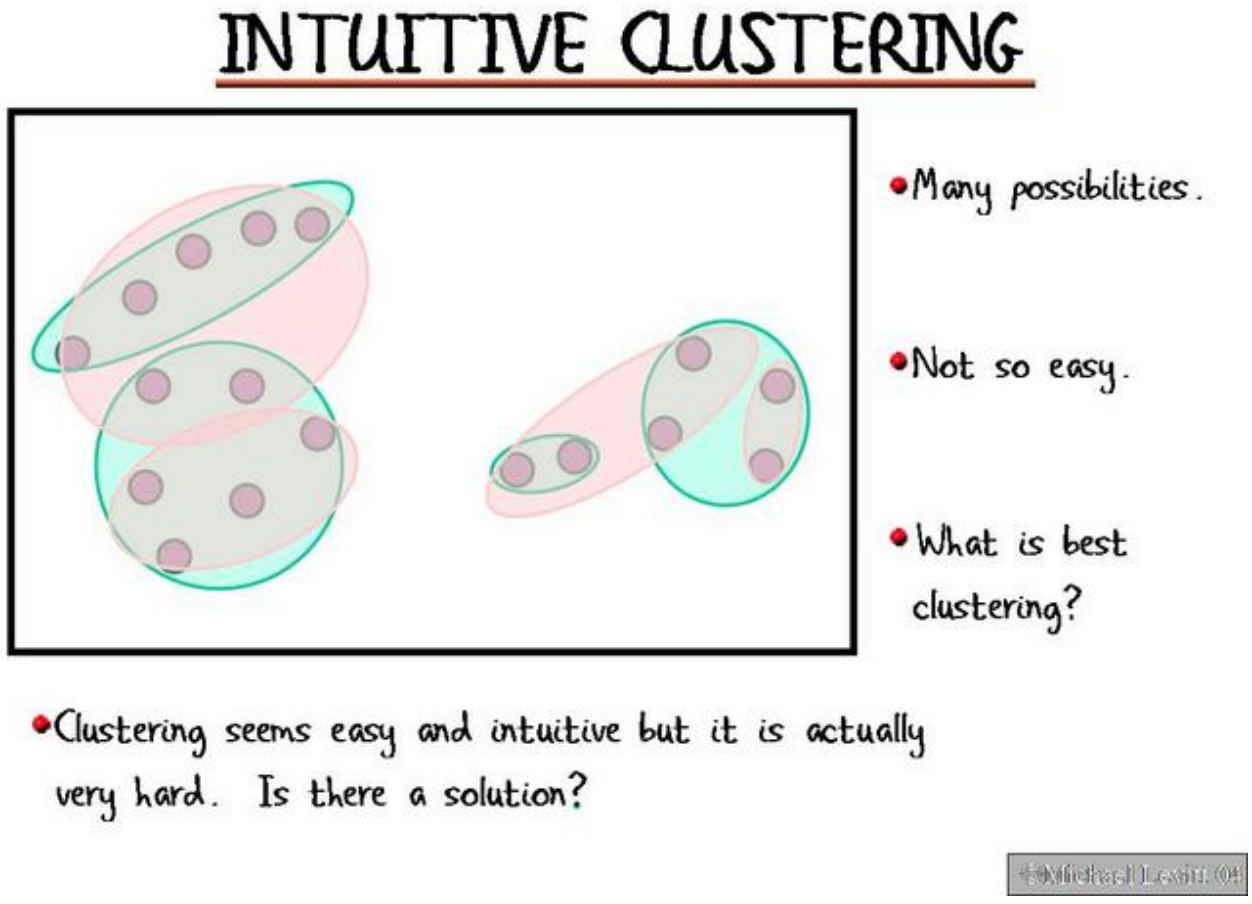
Exercise: search Google images for "cluster analysis" for a zillion examples

Cluster Example 2



[this had something to do with herpetology, perhaps explaining the importance of "road crossings"]

Intuitive Clustering



Diagrams from Michael Levitt, Structural Biology, Stanford
Source:
http://csb.stanford.edu/class/public/lectures/lec4/Lecture6/Data_Visualization/images/Intuitive_Cluste

Overview of distance metrics

Distance Measurements Between Data Points

This parameter specifies how the distance between data points in the clustering input is measured. The options are:

- Euclidean: Use the standard Euclidean (as-the-crow-flies) distance.
- Euclidean Squared: Use the Euclidean squared distance in cases where you would use regular Euclidean distance in Jarvis-Patrick or K-Means clustering.
- Manhattan: Use the Manhattan (city-block) distance.
- Pearson Correlation: Use the Pearson Correlation coefficient to cluster together genes or samples with similar behavior; genes or samples with opposite behavior are assigned to different clusters.
- Pearson Squared: Use the squared Pearson Correlation coefficient to cluster together genes with similar or opposite behaviors (i.e. genes that are highly correlated and those that are highly anti-correlated are clustered together).
- Chebychev: Use Chebychev distance to cluster together genes that do not show dramatic expression differences in any samples; genes with a large expression difference in at least one sample are assigned to different clusters.
- Spearman: Use Spearman Correlation to cluster together genes whose expression profiles have similar shapes or show similar general trends (e.g. increasing expression with time), but whose expression levels may be very different.

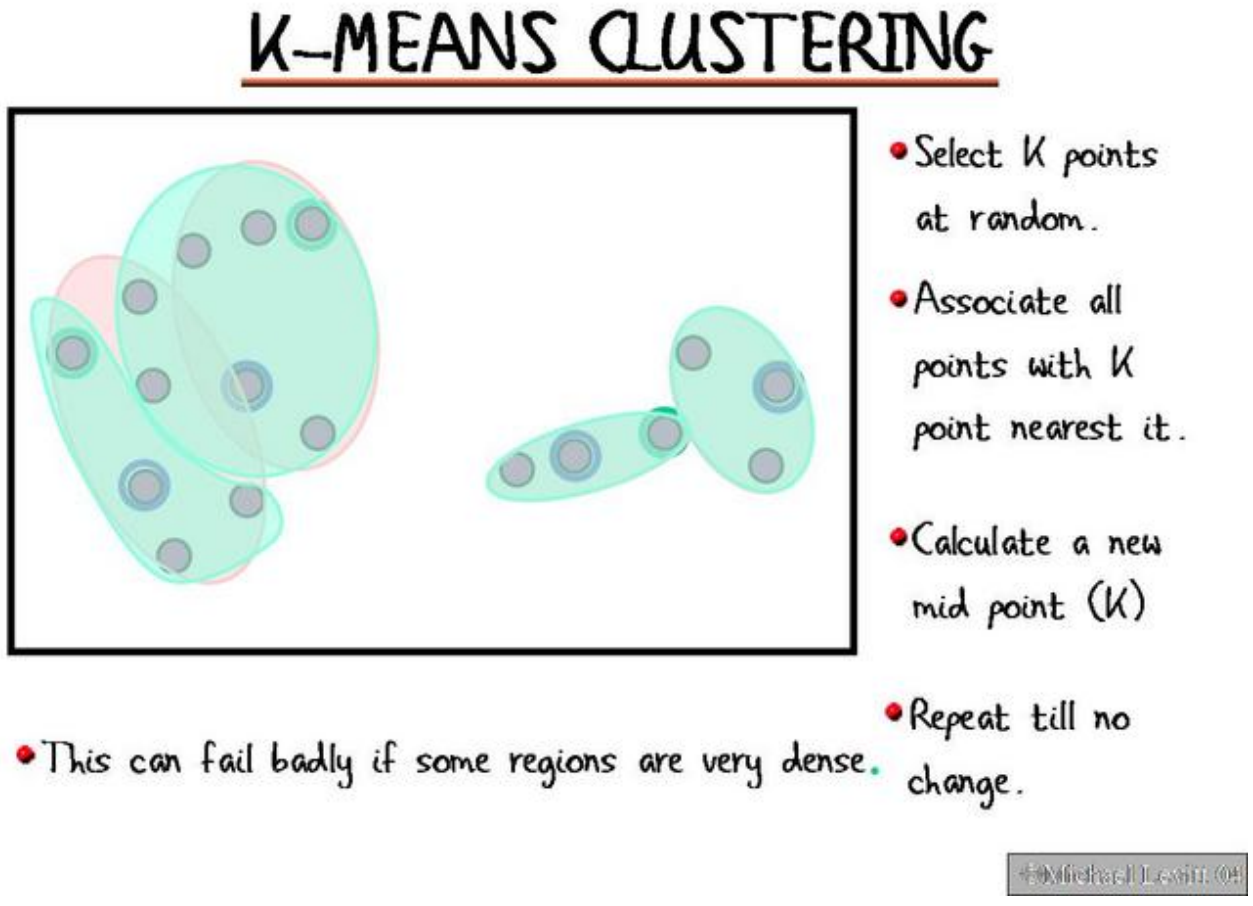
Distance Measurements Between Clusters

This parameter specifies how the distance between clusters is measured. The options are:

- ***Average Linkage***: The distance between two clusters is the average of the distances between all the points in those clusters.
- ***Single Linkage***: The distance between two clusters is the distance between the nearest neighbors in those clusters.
- ***Complete Linkage***: The distance between two clusters is the distance between the furthest points in those clusters.

Source:
http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Distance_Me

K-Means



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

K-Means algorithm

K-means Clustering

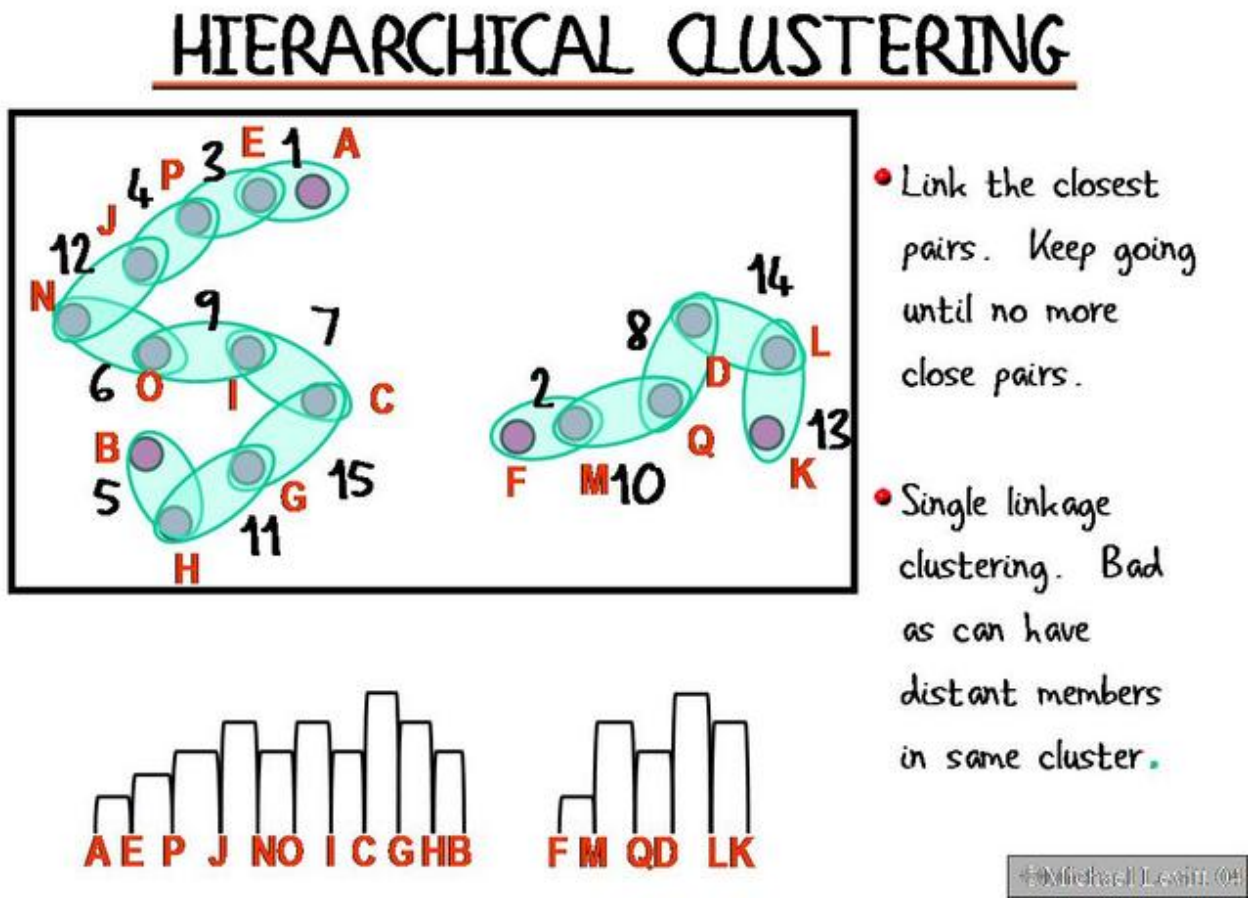
- Randomly assign each of $x_1 \ldots, x_N$ to K user specified clusters
- Compute the average value of the points, or centroid, of each cluster
- For each i=1, ..,N compute the distance between x_i and each of the cluster centroids
- Assign x_i to the cluster with the closest centroid and recalculate the centroids of the affected clusters
- Iterate until no more reassignments are made

Source: http://biology.unm.edu/biology/maggieww/Public_Html/K-means.gif

K-Means: Issues

- Results vary depending on the number of clusters
- Results vary depending on the random starting points: one approach is to do a number of these and see which clusters consistently emerge

Hierarchical Clustering

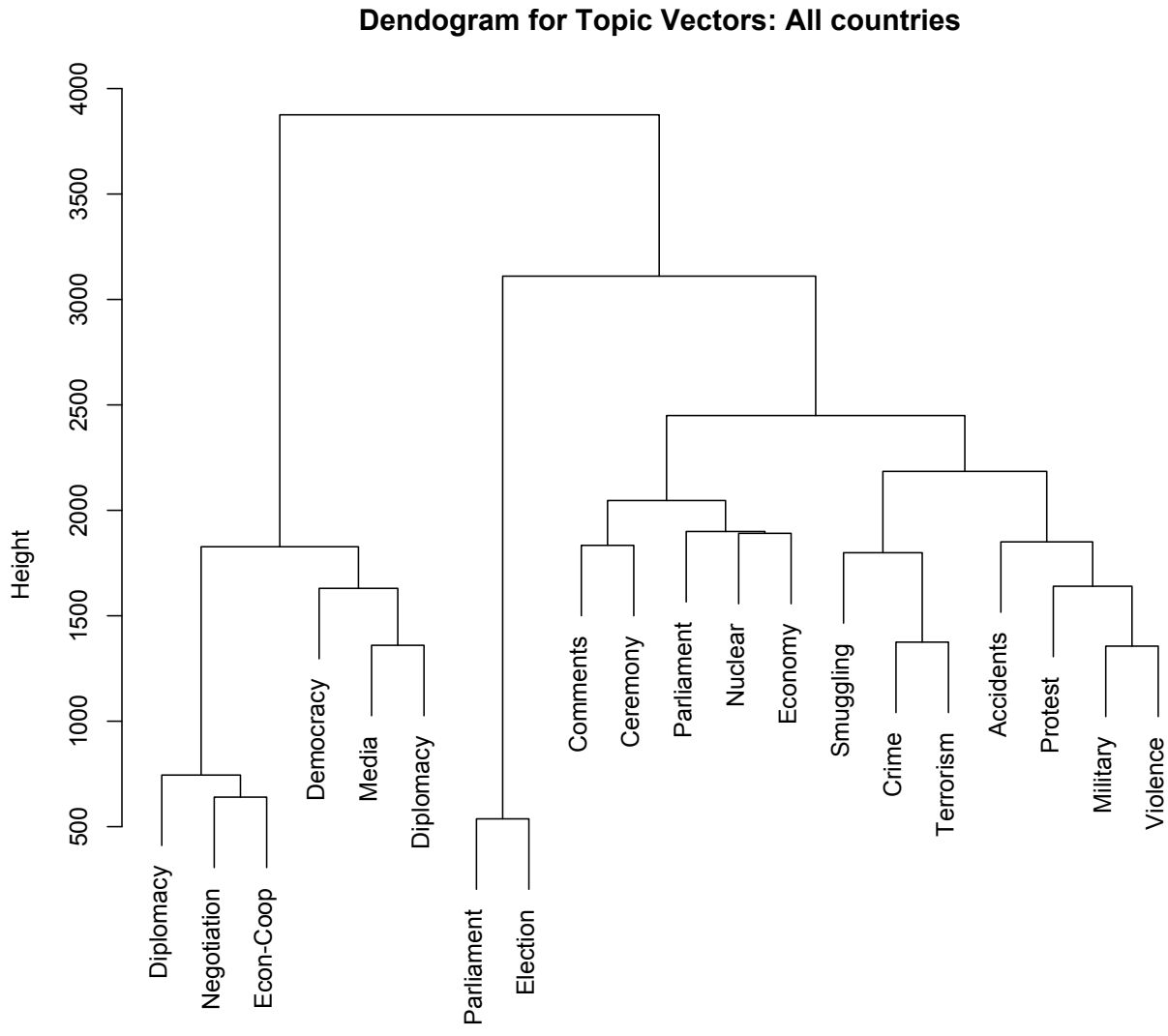


Source:
http://csb.stanford.edu/class/public/lectures/lec4/Lecture6/Data_Visualization/images/Hierarchical_Cl

Comparison Strategy

- Words that are similar should co-occur in topics more frequently
- For a pair of 'top-words', let their similarity-weight be equal to:
  - No. of times that the pair appears *within* all 'top-word' vectors
- Distance between two vectors:
  - A constant minus the sum of the similarity-weights for word-pairs that occur *across* the two 'top-word' vectors

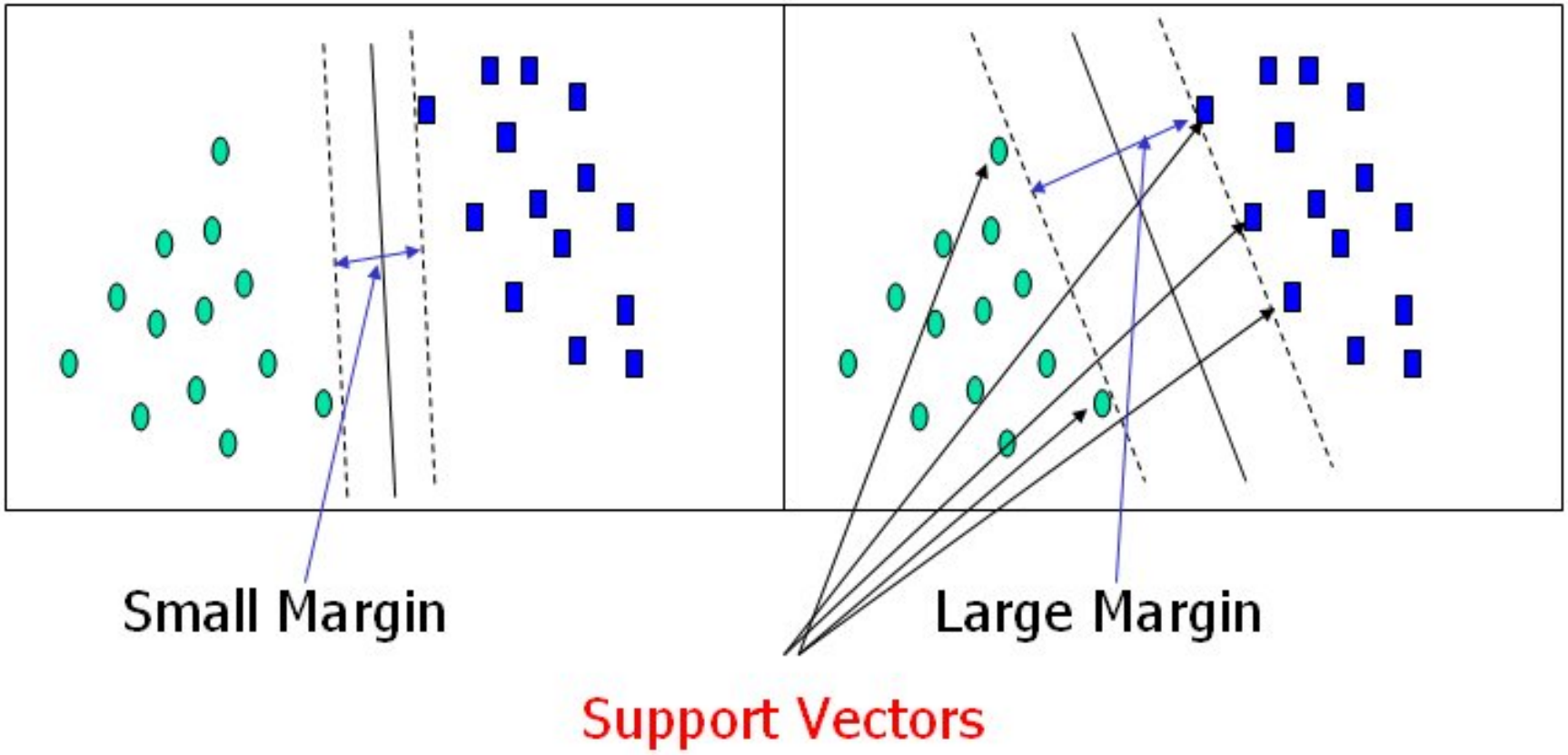
Comparing Topics: Combined Sample



Support Vector Machines

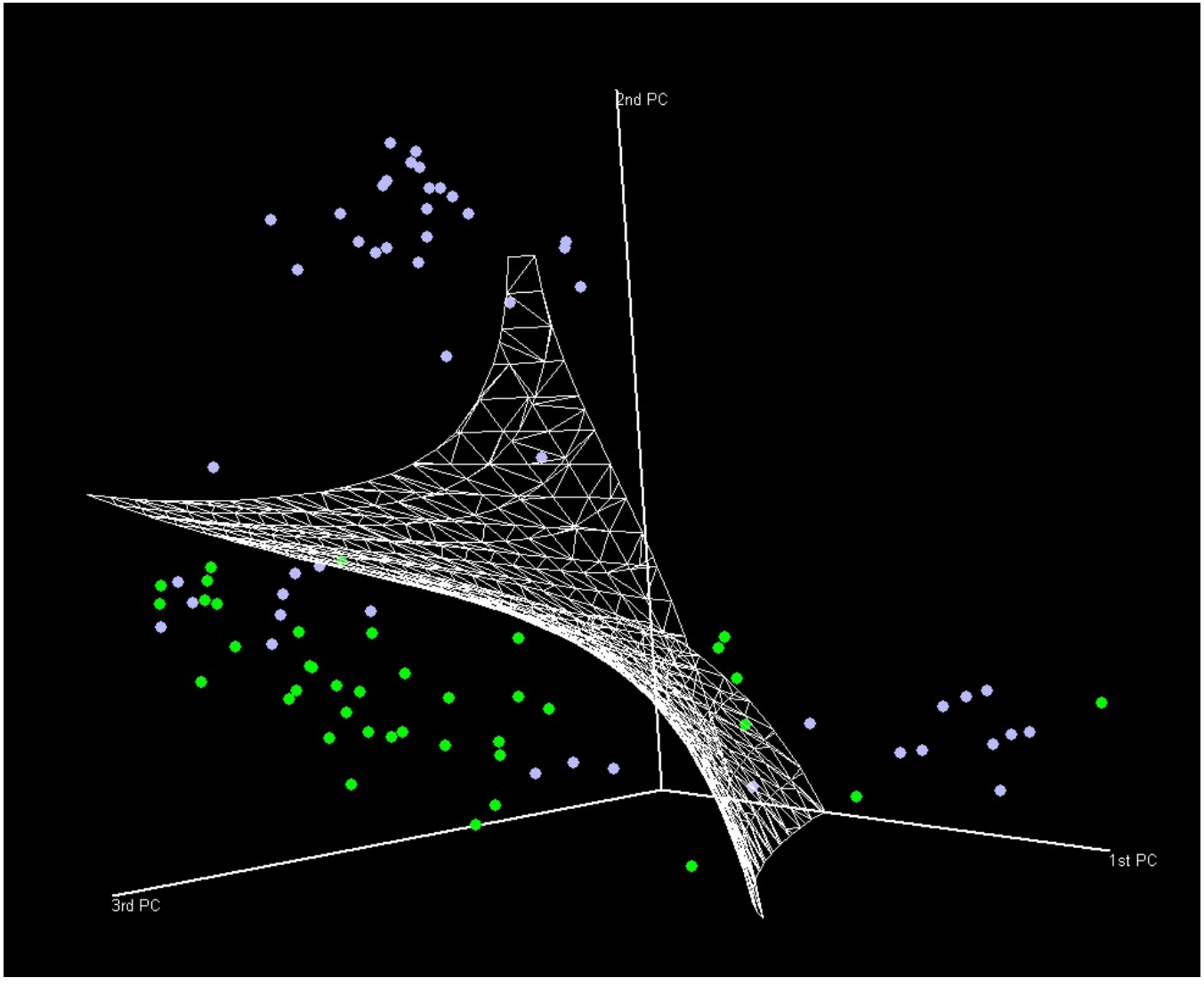
- The support-vector machine (SVM) is the workhorse of document classification and has proven to be highly robust across a wide variety of domains.
- SVM partitions a high-dimensional space into two categories while maximizing the distance between the cases at the boundary
- SVM reduces the number of "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



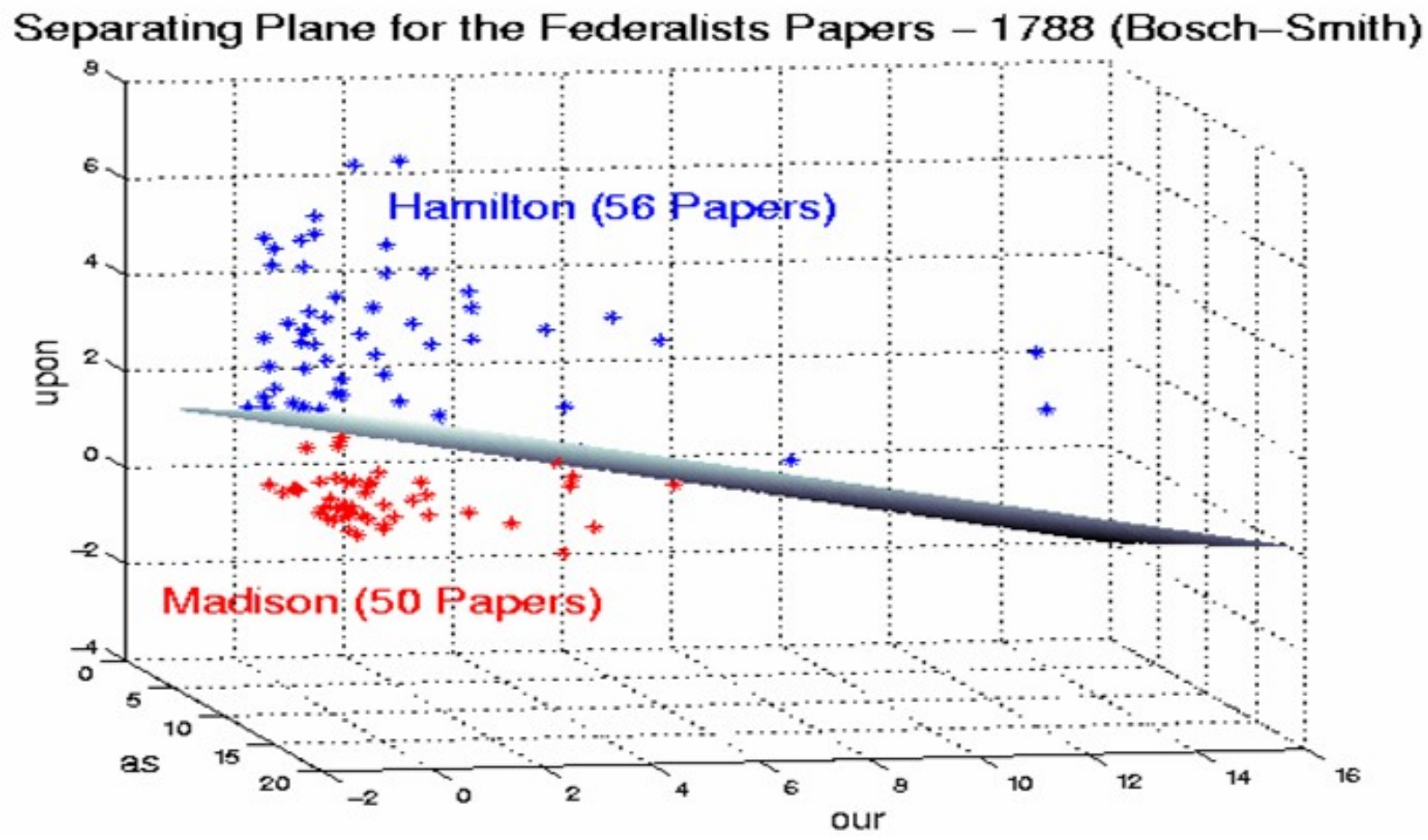
Source: http://www.dtreg.com/svm.htm

A fancier SVM



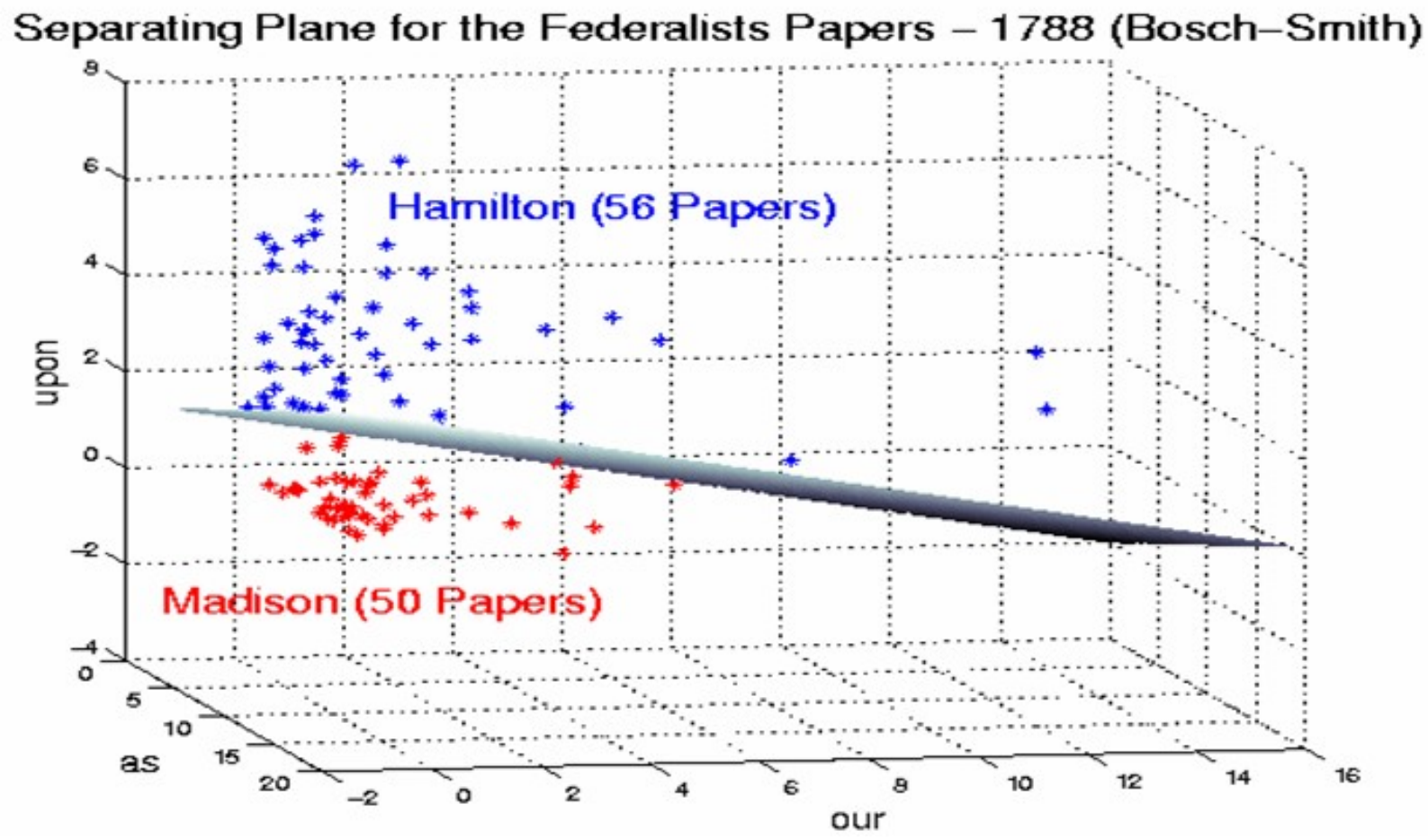
Source: http://www.epicentersoftware.com/genetrix/features/machine_learning_heuristics.htm

Applied SVM



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

Applied SVM

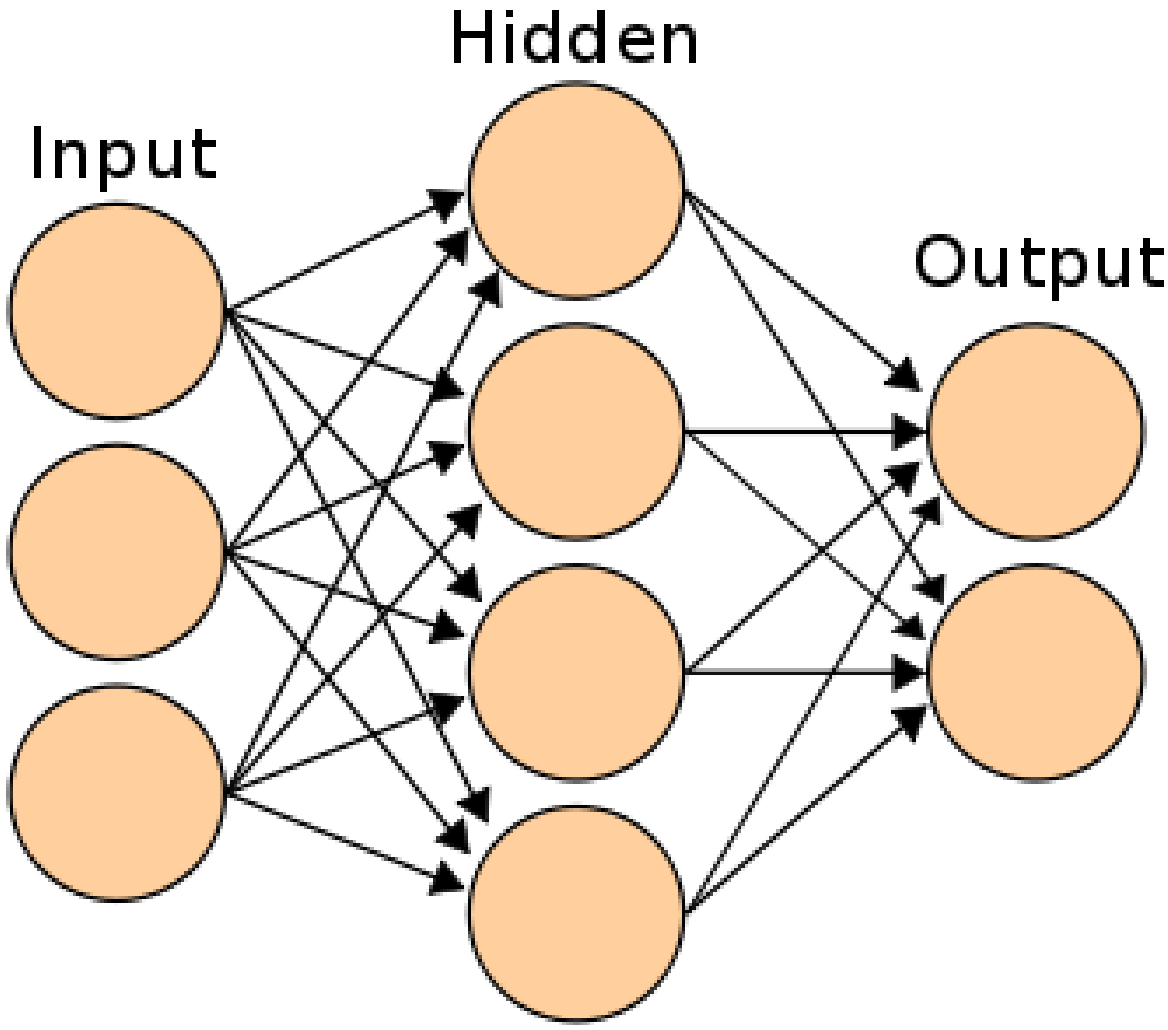


Source: http://www.dtreg.com/svm.htm

Neural networks

Developed by Geoffrey Hinton, who through the magic of the internet, is here to explain...
https://www.coursera.org/course/neuralnets

Neural network



Source: http://www.morosanmihail.com/home/post/javascript-neural-network

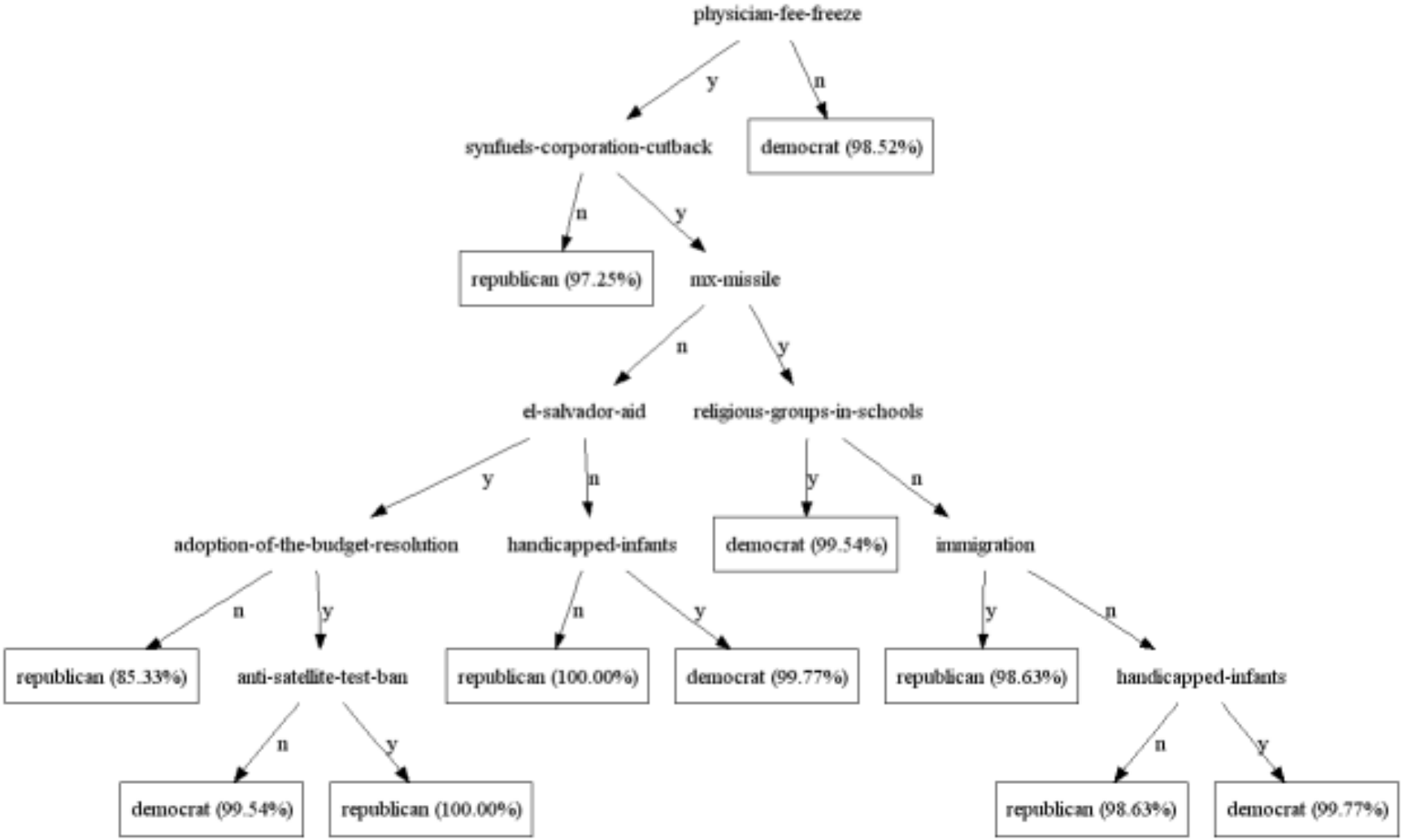
Just one correspondence analysis graphic

since I think the method is cool...

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

Classification Trees

Classification Tree Example



Source: http://orange.biolab.si/doc/ofb/c_otherclass.htm

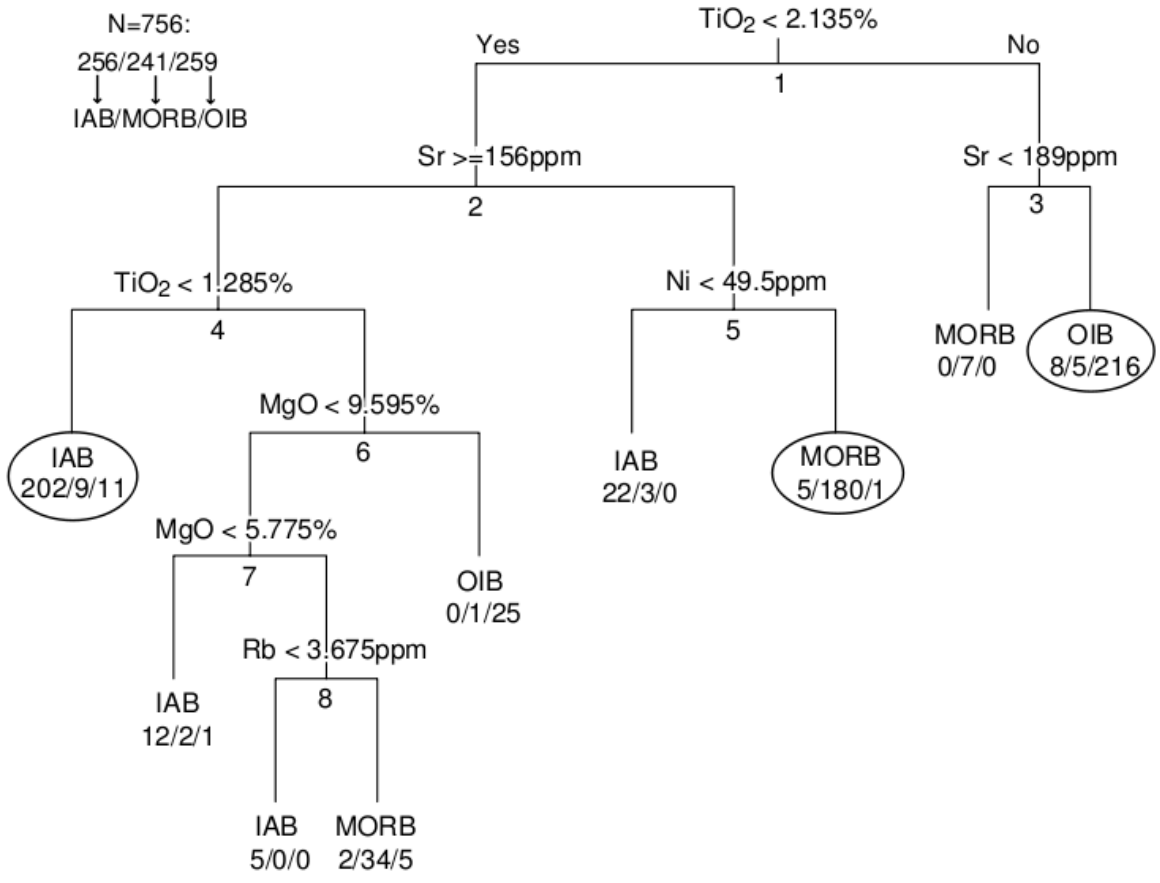
Classification Tree Example

output of running the tree.py script

```
physician-fee-freeze=n: democrat (98.52%)
physician-fee-freeze=y
|    synfuels-corporation-cutback=n: republican (97.25%)
|    synfuels-corporation-cutback=y
|    |    mx-missile=n
|    |    |    el-salvador-aid=y
|    |    |    |    adoption-of-the-budget-resolution=n: republican (85.33%)
|    |    |    |    adoption-of-the-budget-resolution=y
|    |    |    |    |    anti-satellite-test-ban=n: democrat (99.54%)
|    |    |    |    |    anti-satellite-test-ban=y: republican (100.00%)
|    |    |    el-salvador-aid=n
|    |    |    |    handicapped-infants=n: republican (100.00%)
|    |    |    |    handicapped-infants=y: democrat (99.77%)
|    |    mx-missile=y
|    |    |    religious-groups-in-schools=y: democrat (99.54%)
|    |    |    religious-groups-in-schools=n
|    |    |    |    immigration=y: republican (98.63%)
|    |    |    |    immigration=n
|    |    |    |    |    handicapped-infants=n: republican (98.63%)
|    |    |    |    |    handicapped-infants=y: democrat (99.77%)
```

Source: http://orange.biolab.si/doc/ofb/c_otherclass.htm

Classification Tree with Continuous Breakpoints



[this has something to do with classifying basalts]

Source: http://www.ucl.ac.uk/∼ucfbpve/papers/VermeeschGCA2006/W3441-rev37x.png

ID3 Algorithm

- Calculate the entropy of every attribute using the data set S
- Split the set S into subsets using the attribute for which entropy is minimum (or, equivalently, information gain is maximum)
- Make a decision tree node containing that attribute
- Recurse on subsets using remaining attributes

Source: http://en.wikipedia.org/wiki/ID3_algorithm

Entropy: definition

Definition [edit]

Named after Boltzmann's H-theorem, Shannon denoted the entropy H of a discrete random variable X with possible values $\{x_1, ..., x_n\}$ and probability mass function $P(X)$ as,

$$H(X) = E[I(X)] = E[-\ln(P(X))].$$

Here E is the expected value operator, and I is the information content of X.[8][9] $I(X)$ is itself a random variable.

When taken from a finite sample, the entropy can explicitly be written as

$$H(X) = \sum_i P(x_i)\, I(x_i) = -\sum_i P(x_i) \log_b P(x_i) = -\sum_i \frac{n_i}{N} \log_b \frac{n_i}{N} = \log_b N - \frac{1}{N} \sum_i n_i \log_b n_i,$$

where b is the base of the logarithm used. Common values of b are 2, Euler's number e, and 10, and the unit of entropy is bit for $b = 2$, nat for $b = e$, and dit (or digit) for $b = 10$.[10]

In the case of $p(x_i) = 0$ for some i, the value of the corresponding summand $0 \log_b(0)$ is taken to be 0, which is consistent with the well-known limit:

$$\lim_{p \to 0+} p \log(p) = 0.$$

Source: http://en.wikipedia.org/wiki/Entropy_%28information_theory%29

C4.5 Algorithm

C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set $S = s_1, s_2, ...$ of already classified samples. Each sample s_i consists of a p-dimensional vector $(x_{1,i}, x_{2,i}, ..., x_{p,i})$, where the x_j represent attributes or features of the sample, as well as the class in which s_i falls.
At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurses on the smaller sublists.
Source: http://en.wikipedia.org/wiki/C4.5_algorithm

C4.5 vs. ID3

C4.5 made a number of improvements to ID3. Some of these are:

- Handling both continuous and discrete attributes: In order to handle continuous attributes, C4.5 creates a threshold and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it.

- Handling training data with missing attribute values—C4.5 allows attribute values to be marked as ? for missing. Missing attribute values are simply not used in gain and entropy calculations.

- Handling attributes with differing costs.

- Pruning trees after creation—C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes

Random Forests^{TM} : Breiman's Algorithm

Each tree is constructed using the following algorithm:

1. Let the number of training cases be N, and the number of variables in the classifier be M.
2. We are told the number m of input variables to be used to determine the decision at a node of the tree; m should be much less than M.
3. Choose a training set for this tree by choosing n times with replacement from all N available training cases (i.e., take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes.
4. For each node of the tree, randomly choose m variables on which to base the decision at that node. Calculate the best split based on these m variables in the training set.
5. Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier).

For prediction a new sample is pushed down the tree. It is assigned the label of the training sample in the terminal node it ends up in. This procedure is iterated over all trees in the ensemble, and the mode vote of all trees is reported as the random forest prediction.

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

This sucker is trade-marked!

Random Forests(tm) is a trademark of Leo Breiman and Adele Cutler and is licensed exclusively to Salford Systems for the commercial release of the software. Our trademarks also include RF(tm), RandomForests(tm), RandomForest(tm) and Random Forest(tm).

For details:
http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

Features of Random Forests

Breiman et al claim the following:

- It is unexcelled in accuracy among current algorithms.
- It runs efficiently on large data bases.
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
- It has methods for balancing error in class population unbalanced data sets.
- Generated forests can be saved for future use on other data.
- Prototypes are computed that give information about the relation between the variables and the classification.
- It computes proximities between pairs of cases that can be used in clustering, locating outliers, or (by scaling) give interesting views of the data.
- The capabilities of the above can be extended to unlabeled data, leading to unsupervised clustering, data views and outlier detection.
- It offers an experimental method for detecting variable interactions.

Random forests^{TM} may also cure acne, remove cat hair from upholstery and show promise for bringing peace to the Middle East, though Breiman et al do not explicitly make these claims.

Source: http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#features

Bayesian Model Averaging

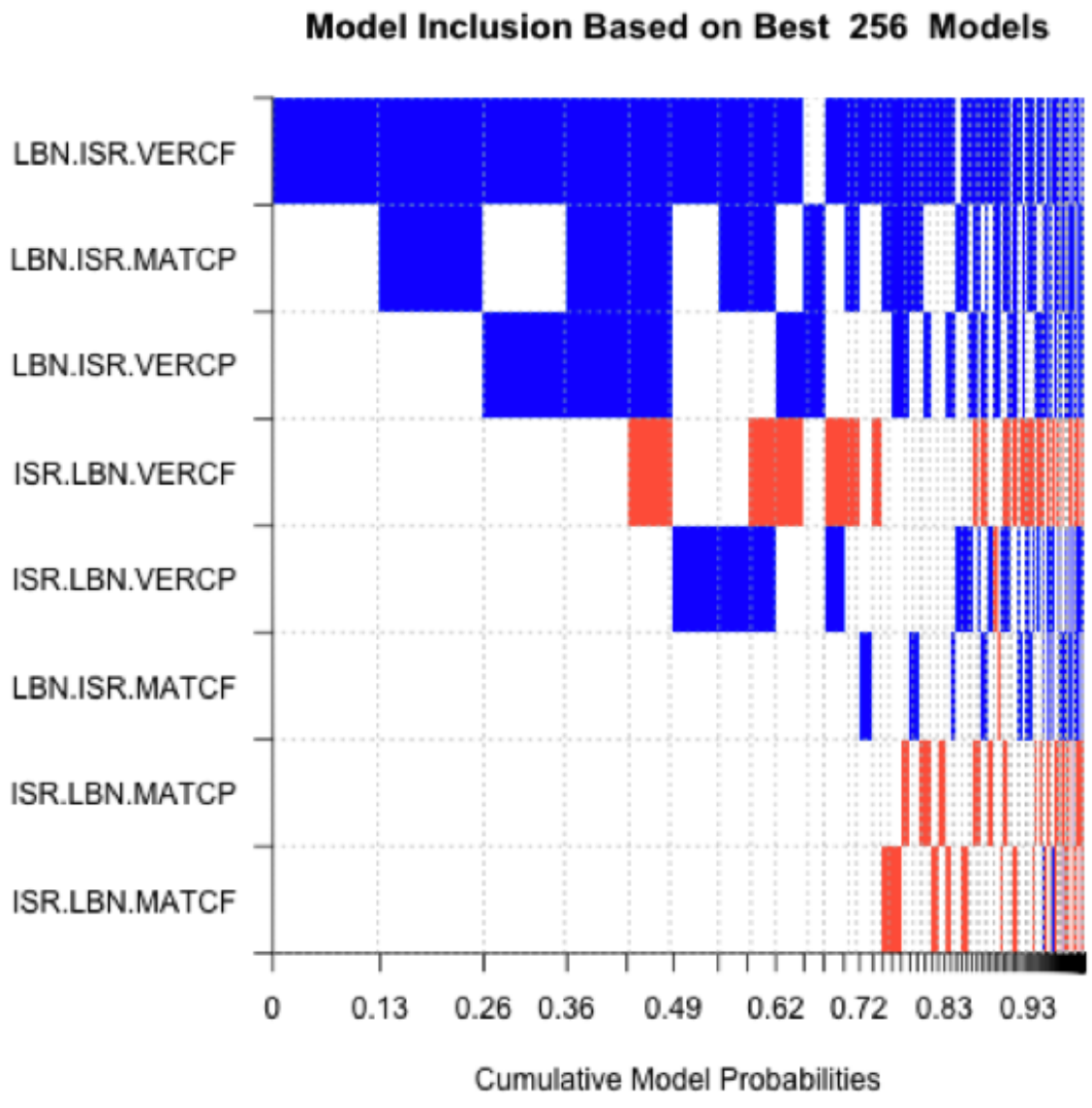
- Systematically integrates the information provided by all combinations of variables
- Result is the overall posterior probability that a variable is important
  - Without having to generate hundreds of papers and thousands of non-randomly discarded models
- Machine learning suggests that systematic assessment of models gives about 10% better accuracy with much less information, and completely eliminates the need for vaguely defined indicators
- Predictions can be made using an ensemble of all of the models
  - In meteorology and finance, these models are generally more robust in out-of-sample evaluations
- Framework is Bayesian rather than frequentist, which eliminates a long list of philosophical and interpretive problems with the frequentist approach

The problem of "controls"

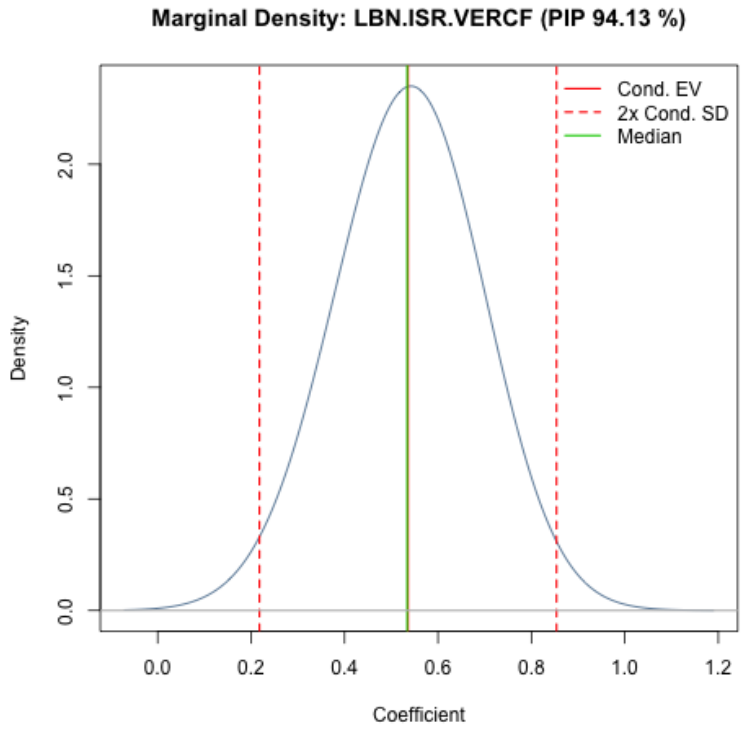
- For starters, they aren't "controls", they are just another variable
  - Often in a really bad [colinear] neighborhood
  - Nature bats last in $(X'X)^{-1}X'y$
  - For something closer to a control, use case matching or Bayesian priors
- Numerous studies over the past 50 years—all ignored (Kahneman)—have suggested that simple models are better
- In many forecasting models, there is no obvious theoretical reason for using any particular measure, so instead we have to assess multiple measures of the same latent concept: "power", "legitimacy", "authoritarianism"
  - This is a feature, not a bug
  - Regression approaches have terrible pathologies in these situations
  - Currently, we laboriously work through all of these options across scores of journal and conference papers presented over the course of years*

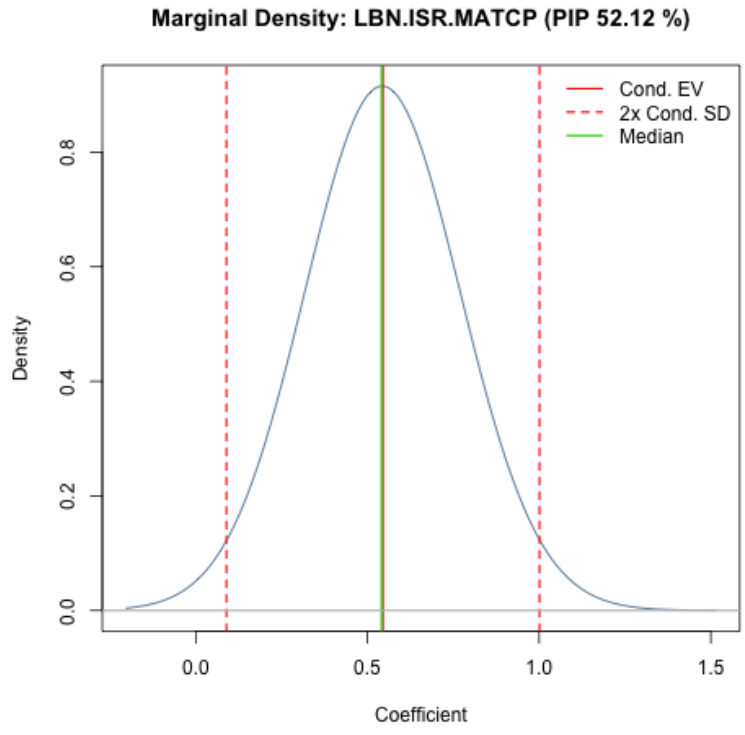
* So if BMA really catches on, a number of journals—and tenure cases—are doomed. On the former, how sad. On the latter, be afraid, be very afraid.

BMA: variable inclusion probabilities



BMA: Posterior probabilities





Sequence models

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

The Lure of Sequence and Trigger Models

- Kahneman/Tetlock: pattern recognition
- Case-based reasoning
- People—as in "me"—have been working with these for about thirty years without getting a lot of traction. However, we may not have had sufficient detail in the past

Upshot: the only way we are going to know if these are real is if we can train a machine to do this. We may not be able to.

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 → sitten (substitution of ‘s’ for ‘k’)
2. sitten → sittin (substitution of ‘i’ for ‘e’)
3. sittin → sitting (insertion of ‘g’ at the end).

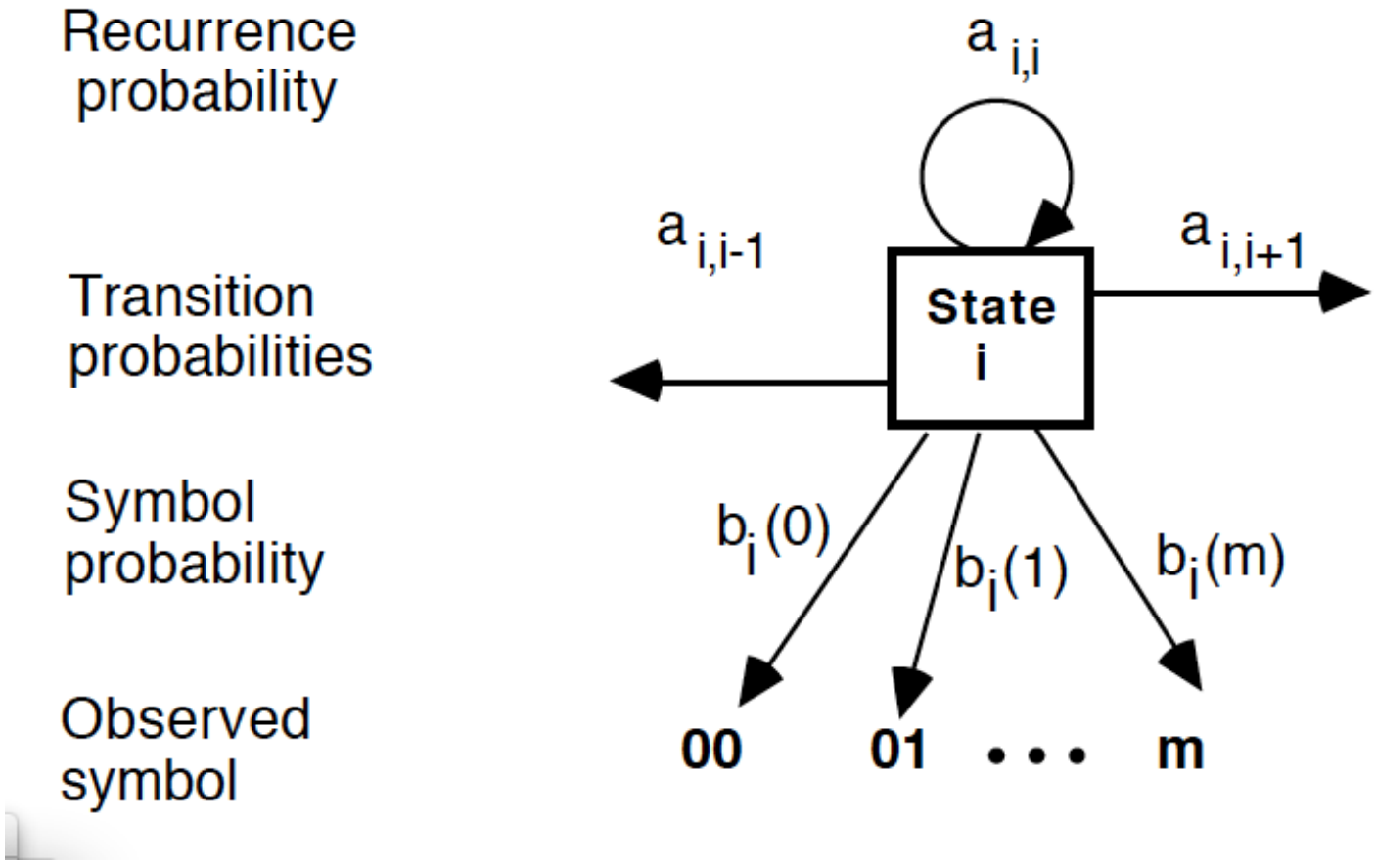
Hidden Markov Model - 1

- 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

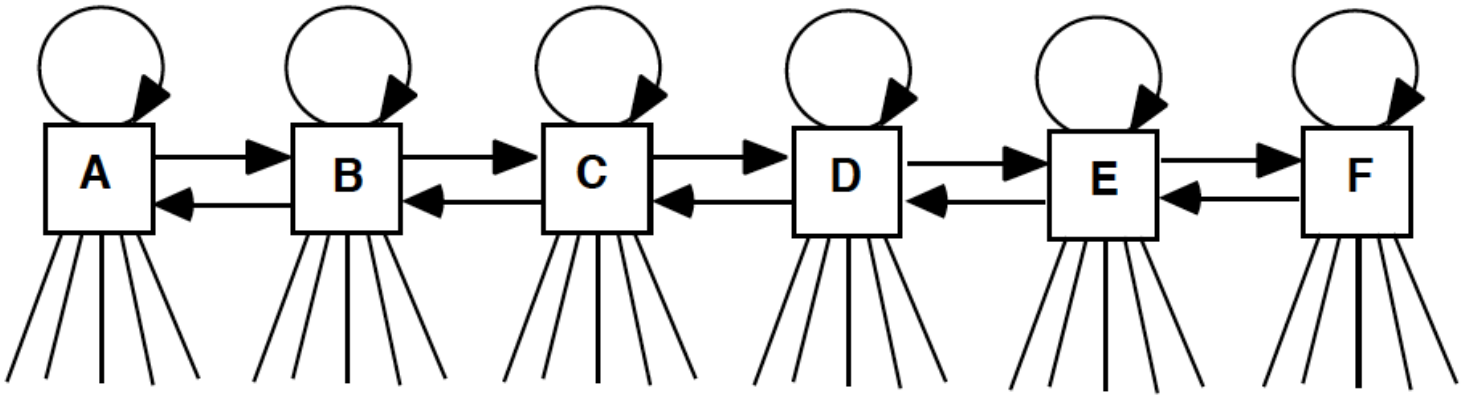
Hidden Markov Model - 2

- 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 lengths

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



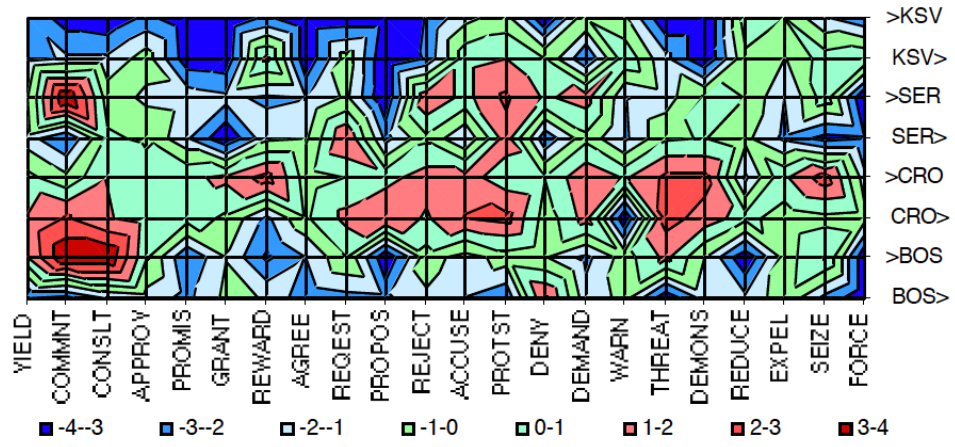
A left-right-left (LRL) hidden Markov Model



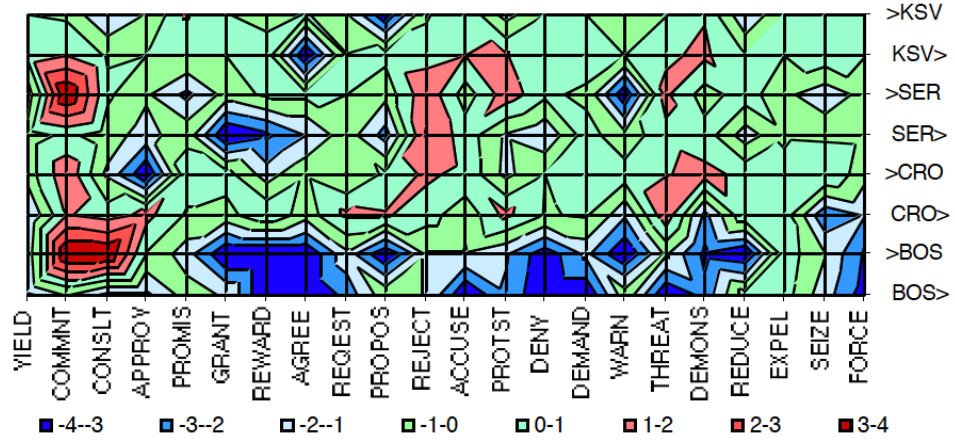
HMM probability map for Balkans

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

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

Conditional Random Fields

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.

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

Problems that bioinformatics do not solve easily

- "Partial ordering" of event sequences—events within a day are randomly ordered—has no analogy in biological sequences
  - Partial ordering problem is less of an issue if one is dealing with aggregated events such as the ICEWS EOIs, since these will almost never occur simultaneously.
- Biological sequences are related through evolutionary change, which provides much closer and systematic matches than those in event sequences
- Biological sequences probably have considerably less variation, even across very different species, than event sequences
  - Though again, this is much less of an issue with macro events
- Noise—non-coding introns—in biological sequences generally occurs in chunks separating contiguous sequences of coding elements
  - "Non-coding" for political events would mostly be situations where there is a "pause" in the crisis: an issue but not a particularly difficult one;

Final thoughts and suggestions

Major lessons learned so far

- There are strong theoretical reasons to believe that error cannot be reduced to zero, but there is no reason why it is stabilizing at 80%
- Successful models are generally relatively simple
- Multiple methods generally converge to similar levels of accuracy, though there are probably minor gains to be made by refining these methods
- Ensemble methods are proving successful for both technical and human forecasts
- Event-based and structural models are *probably* substitutable at relatively short time frames

Some research frontiers that could be productive

- Statistical methods have been explored more thoroughly than machine-learning methods
- Event-based prediction at short horizons—less than 3 months—is largely unexplored
- Sequence-based models are still largely unexplored, though the existing work suggests they are at least credible
- Short-term trigger models may or may not be a hindsight bias illusion: this needs additional work
- At what time horizon and with what pattern do errors occur in high density/long time-series datasets

Thank you

Email: schrodt735@gmail.com

Slides: http://eventdata.parusanalytics.com/presentations.html

Forecasting papers:
http://eventdata.parusanalytics.com/papers.html