Forecasting Conflict
Lecture 5
Machine Learning and Sequence-based Approaches

Philip A. Schrodt

Parus Analytical Systems
schrodt735@gmail.com

Graduate School of Decision Sciences
University of Konstanz
14 - 17 October 2013

Overview

- The Lure of Sequence and Trigger Models
- Cluster-based approaches
  - Logic of nearest neighbor approaches
  - K-Means and its neighbors
  - Dendograms
  - Support vector machines
  - Correspondence analysis
- Other Machine-Learning Methods
  - Classification Trees and Random Forests
  - Genetic Algorithms
  - Neural networks
- Sequence Models
  - Hudson, Schrodt and Widmer: Rule-based approaches
  - Sequence Comparison: Levenshtein Metric
  - Hidden Markov models and Conditional Random Fields

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

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.

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

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)

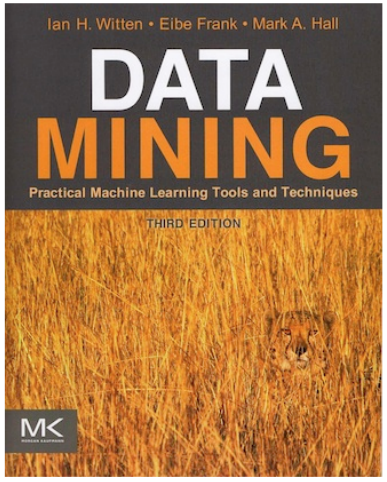
Weka Project: Data Mining

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."

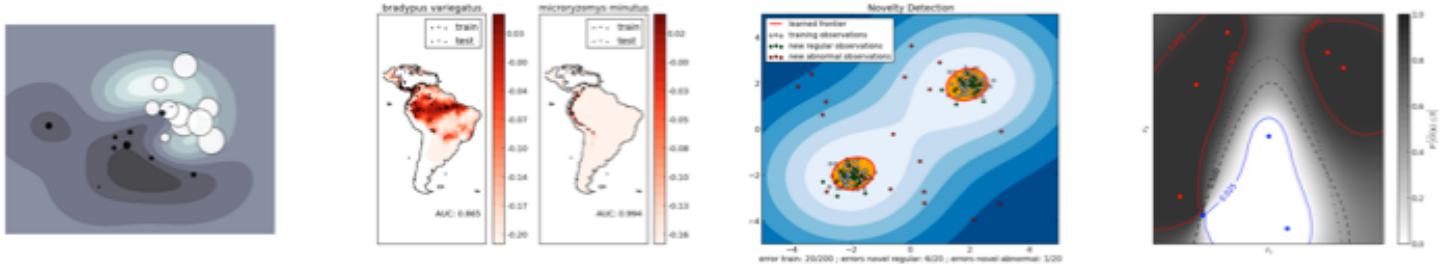
-Herb Edelstein, Principal, Data Mining Consultant, Two Crows Consulting

"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)

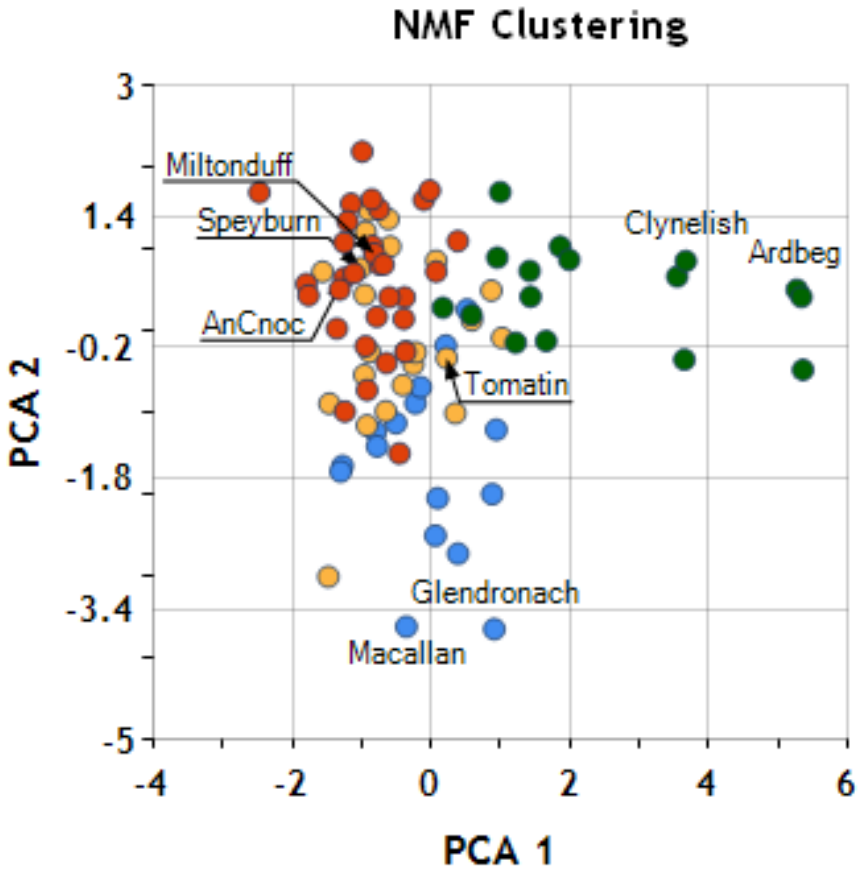
Source: http://scikit-learn.org/stable/

Clustering approaches

General comments

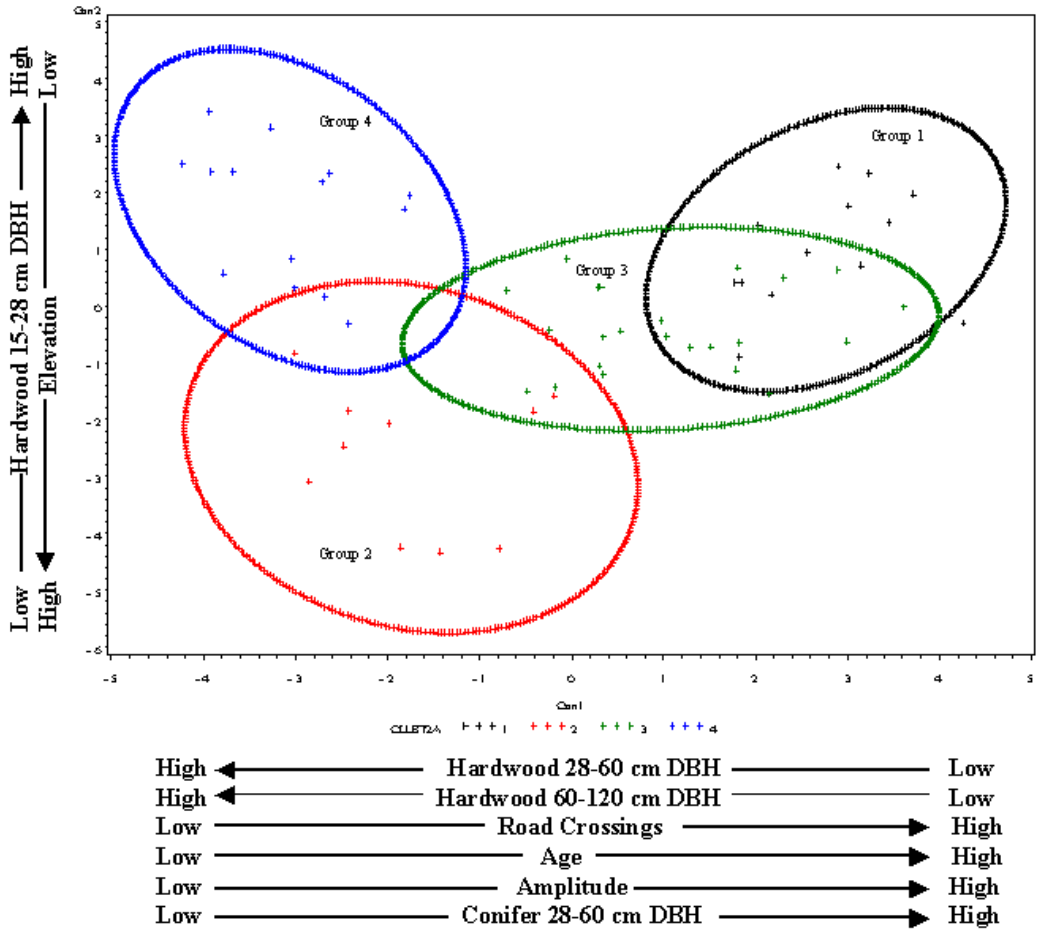
- Requires a metric—and there are many—for the distance between the cases
- In contrast to linear approaches this *assumes* heterogeneous subpopulations
- Clustering is typically depicted in two dimensions but usually is computed in an arbitrarily large space

Cluster Example 1



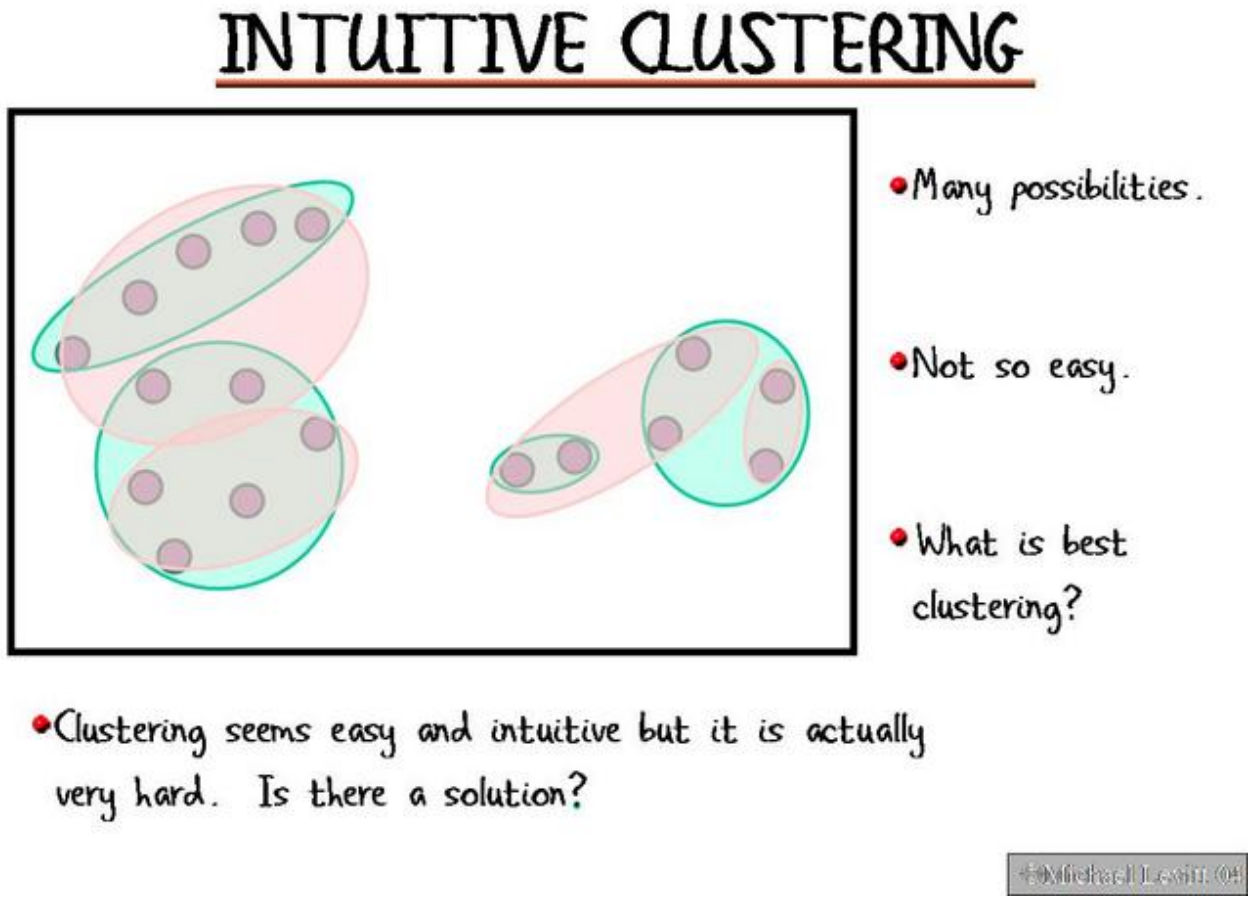
Exercise: search Google images for "cluster analysis" for a zillion examples

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.

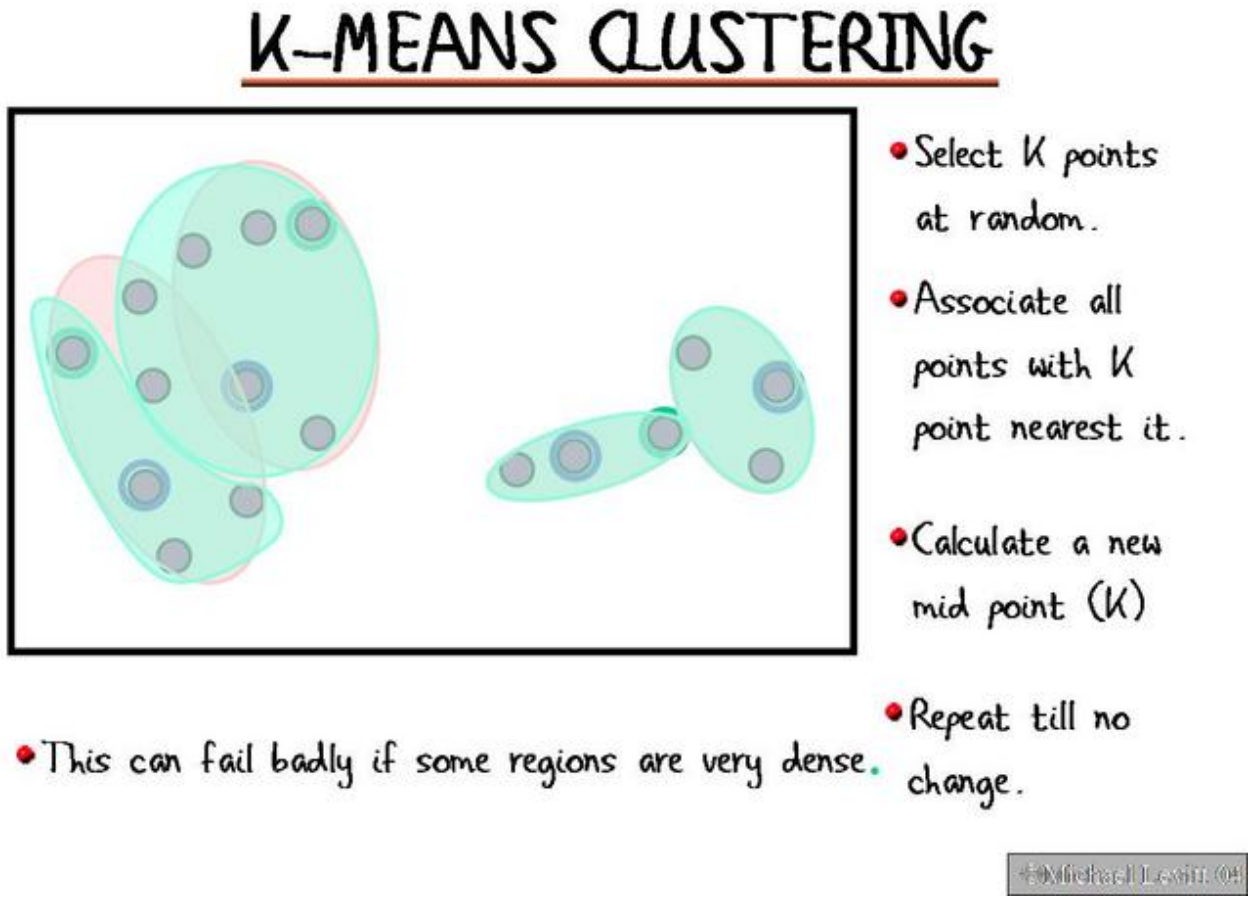
Distance Measurements Between Clusters

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

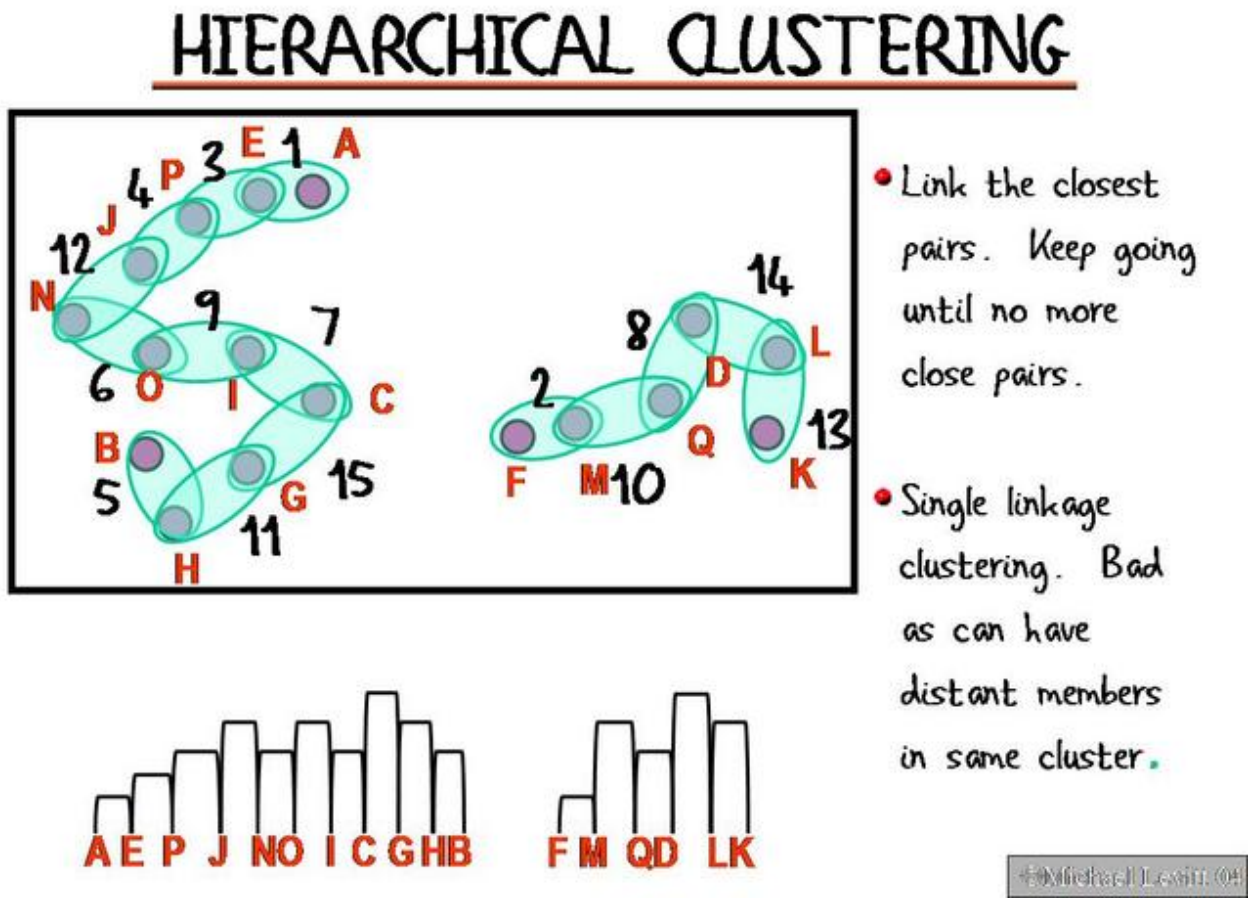
- Randomly assign each of $x_1 \ldots, x_N$ to K user specified clusters
- Compute the average value of the points, or centroid, of each cluster
- For each i=1, ..,N compute the distance between x_i and each of the cluster centroids
- Assign x_i to the cluster with the closest centroid and recalculate 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

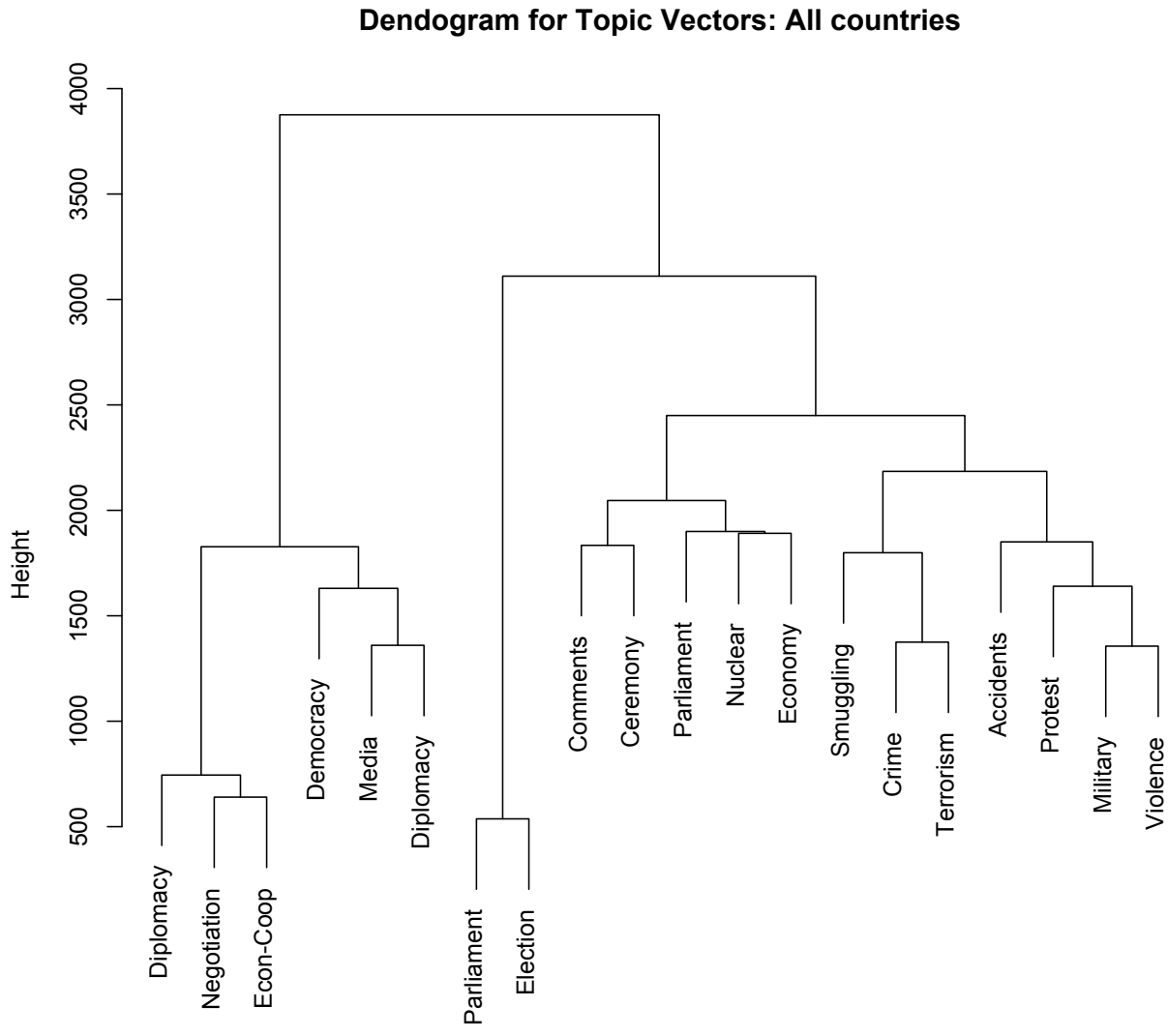


Source:
http://csb.stanford.edu/class/public/lectures/lec4/Lecture6/Data_Visualization/images/Hierarchical_Cl

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

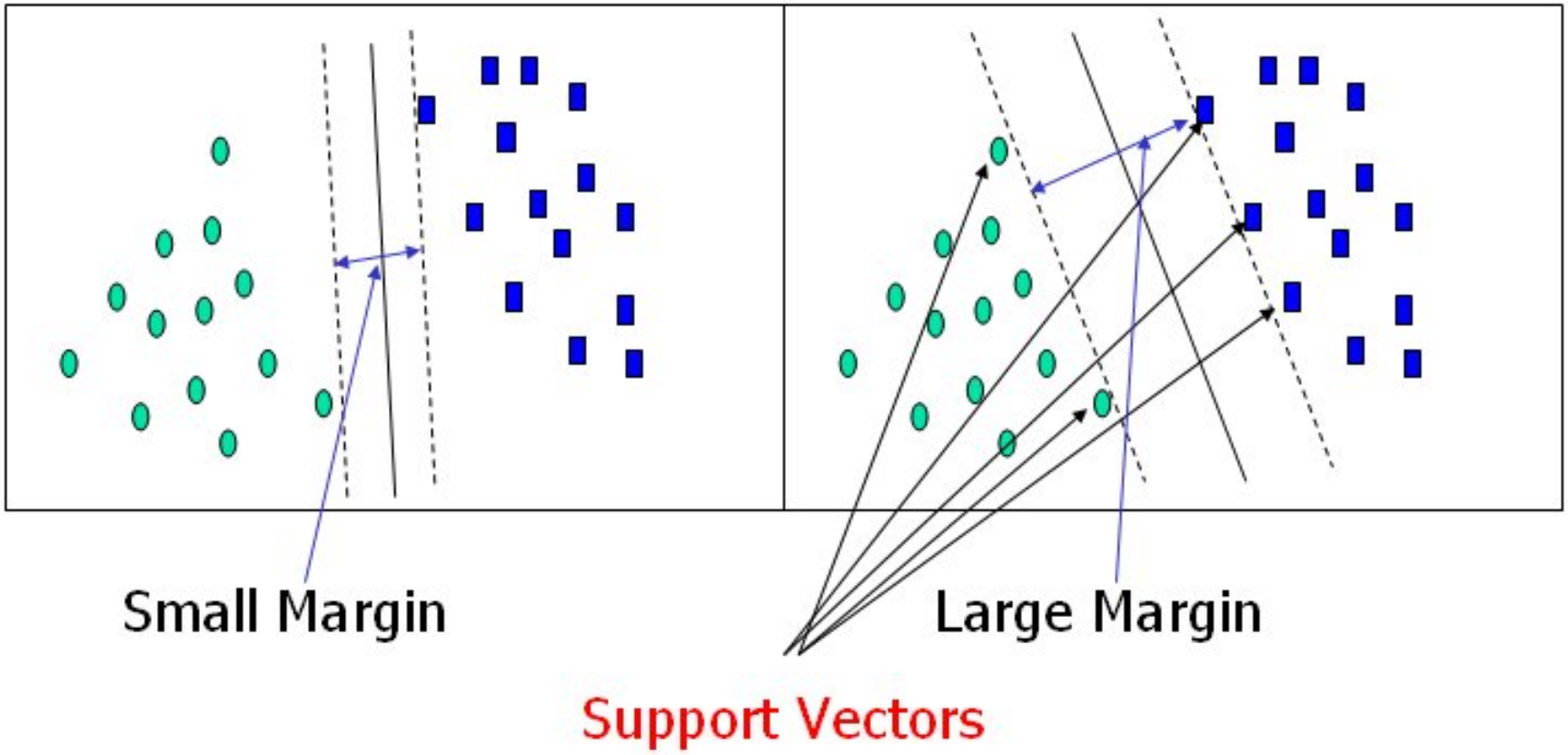
Comparing Topics: Combined Sample



Support Vector Machines

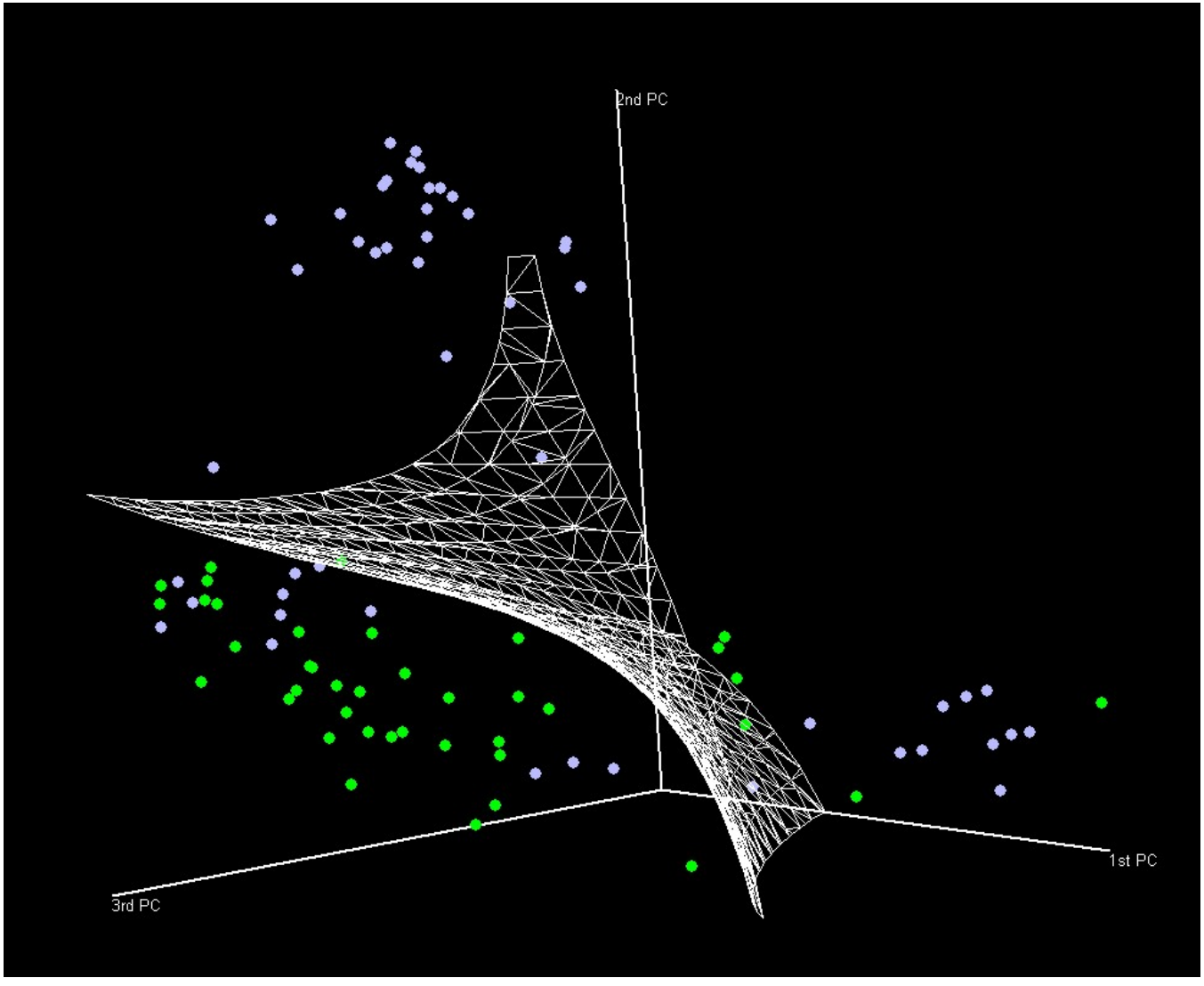
- The support-vector machine (SVM) is the workhorse of document classification and has proven to be highly robust across a wide variety of domains.
- SVM partitions a high-dimensional space into two categories while maximizing the distance between the cases at the boundary
- SVM reduces the number of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



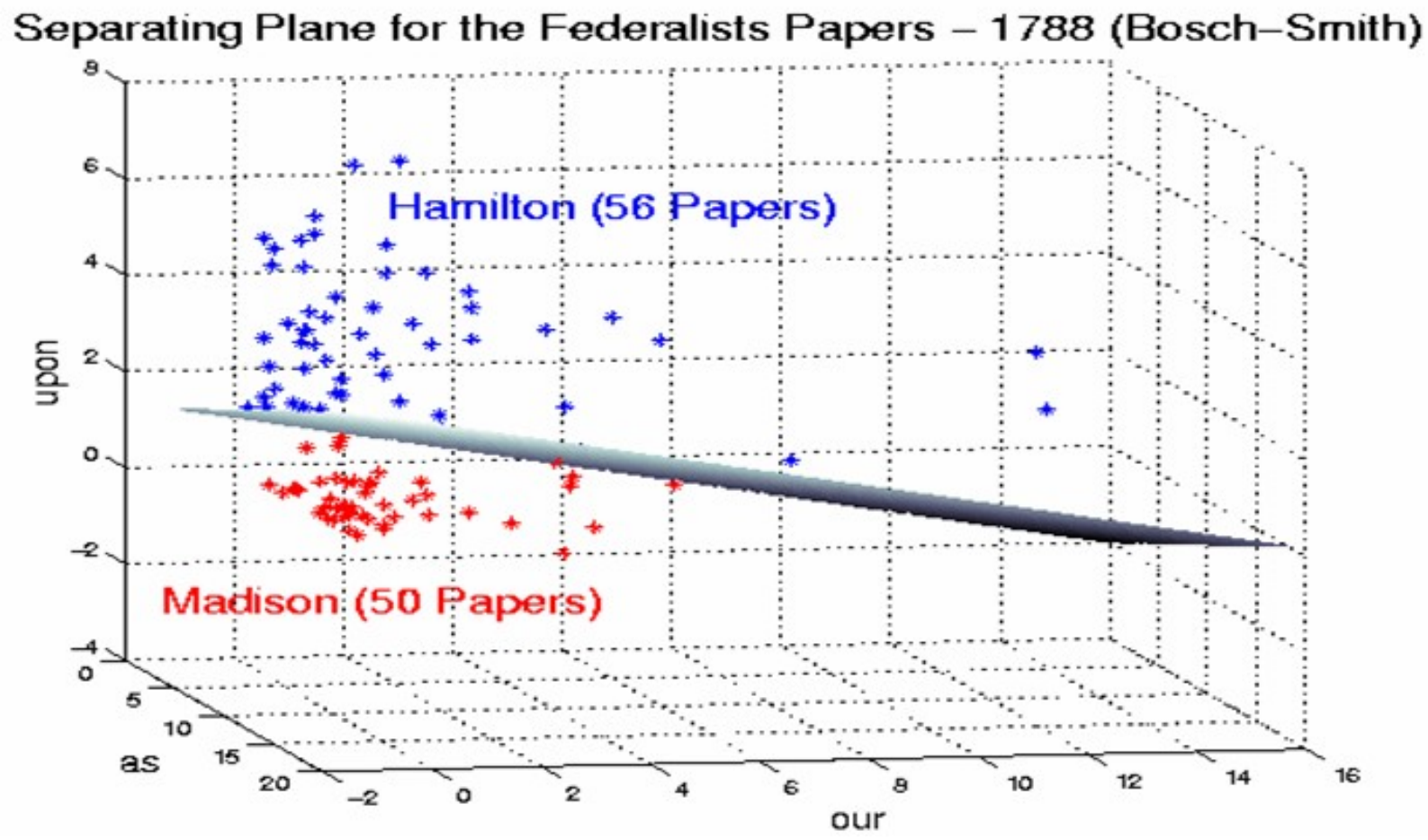
Source: http://www.dtreg.com/svm.htm

A fancier SVM



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

Applied SVM



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

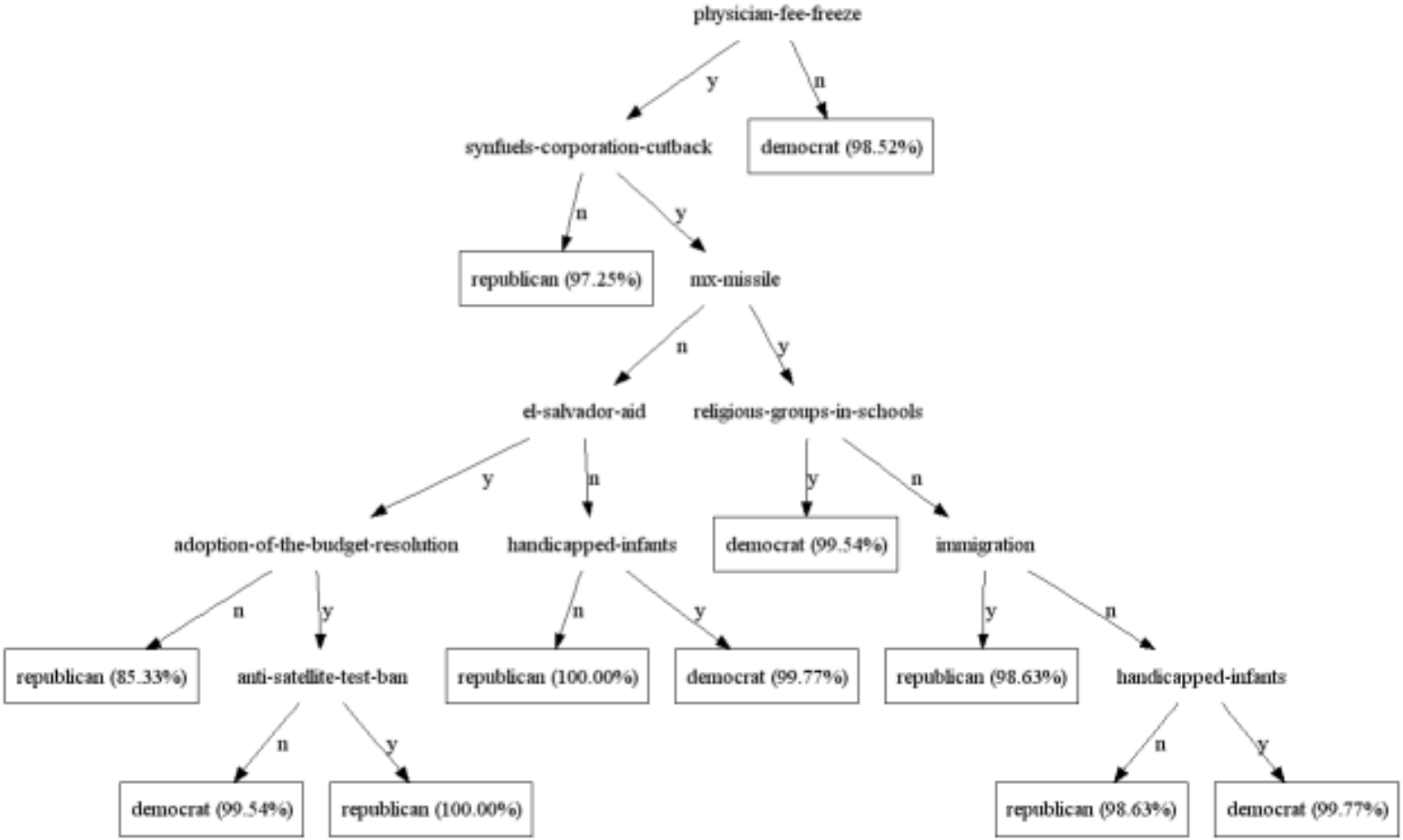
Just one correspondence analysis graphic

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

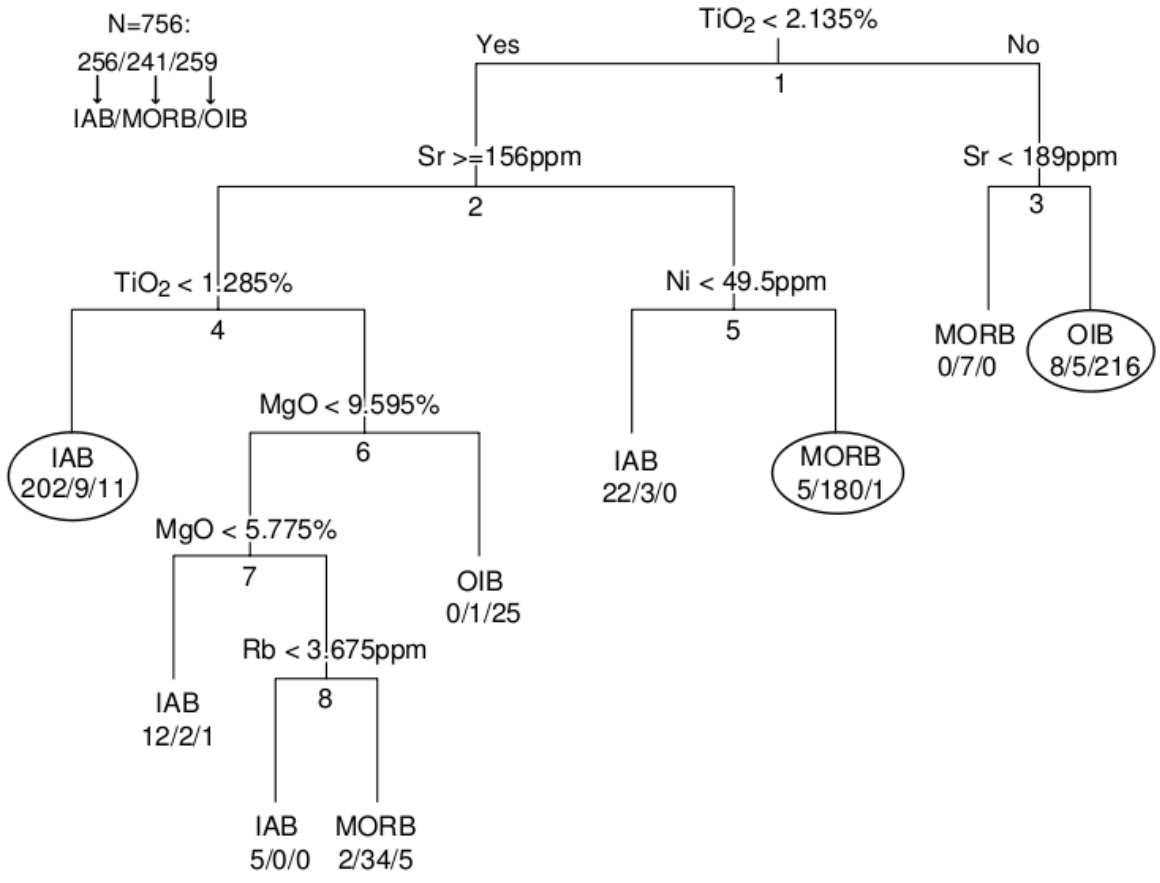
Classification Tree Example

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

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

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).

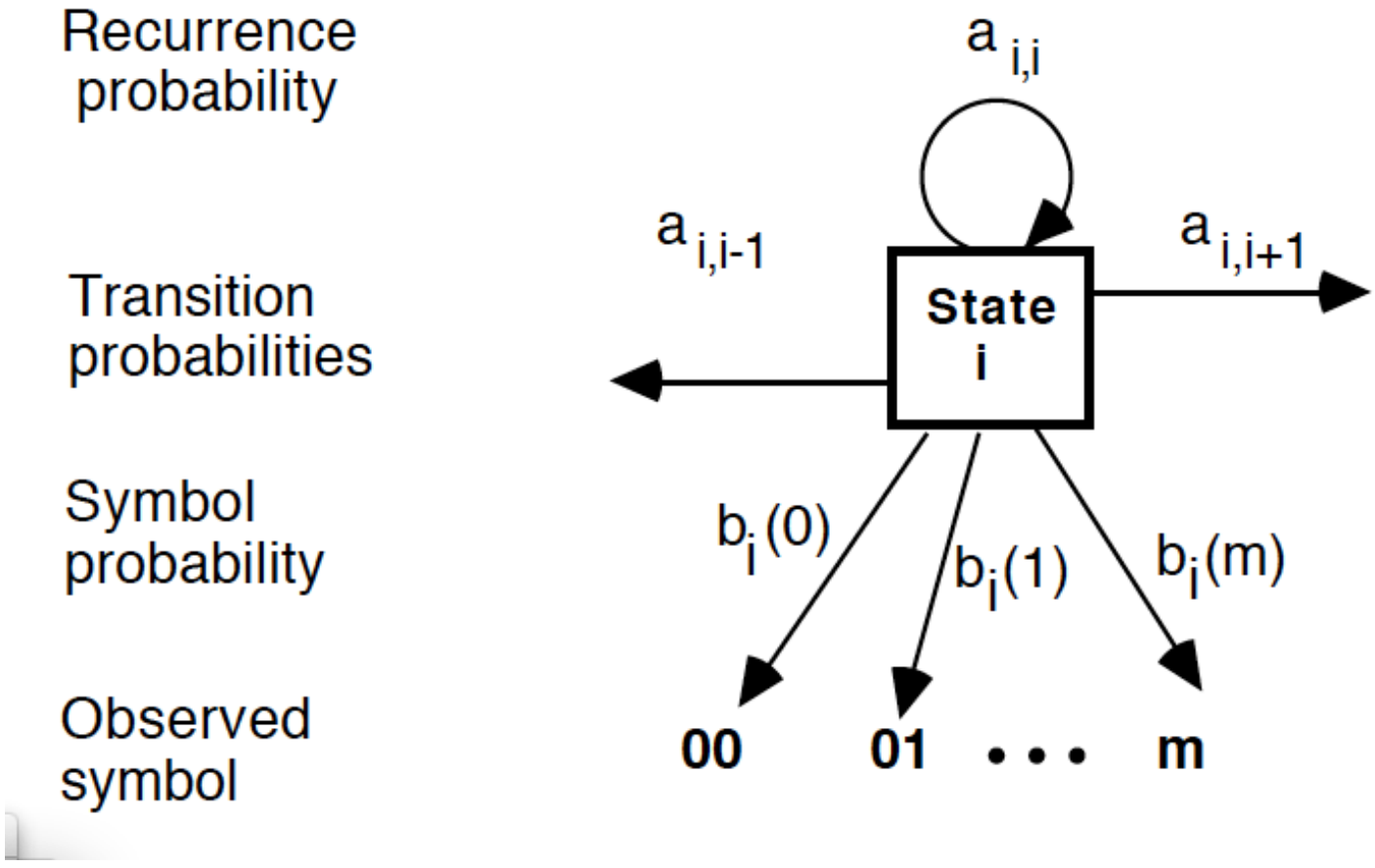
Hidden Markov Model - 1

- 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

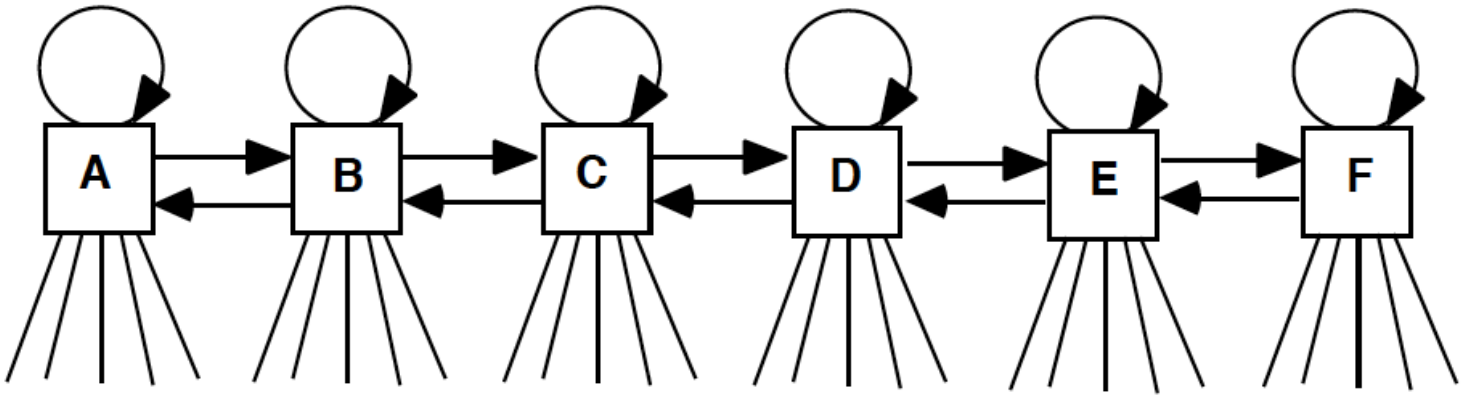
Hidden Markov Model - 2

- 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



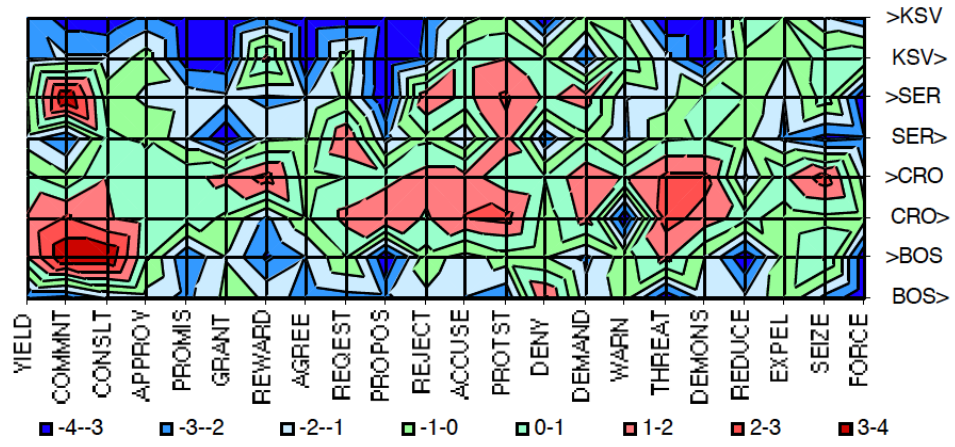
A left-right-left (LRL) hidden Markov Model



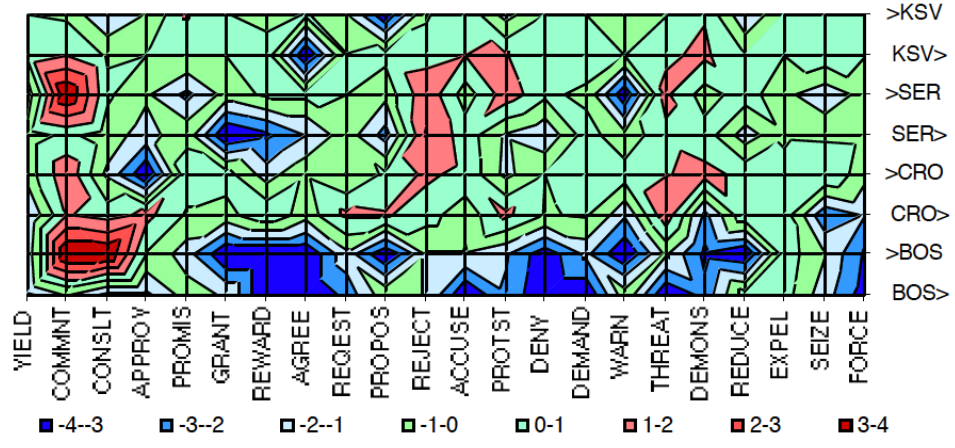
HMM probability map for Balkans

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

Biological sequence comparison

- Massive public investment in North America and Europe on "bioinformatics" tools over the past twenty years.
- Conveniently, versions of most of these are available on the web
- Albeit the background databases have biological rather than political sequences
- More "advanced" commercial versions are also available if this approach pans out, but they are expensive
- There is some overlap between these and existing political sequence analysis approaches, particularly HMMs and CRFs

A convenient set of coincidences...

- DNA has four bases
- The standard CAMEO treatment of events uses four primary categories; verbal/material conflict/cooperation

- Proteins are constructed of 20 amino acids
- CAMEO contains 20 cue categories

- Upshot (and, honest, this was coincidence): one could use the existing DNA and protein sequence analysis software without modification

Characteristics of bioinformatics sequence comparison algorithms

- Provide a variety of comparison metrics involving fixed sequences of a finite set of elements
- Assume random insertion/deletion/mutation of elements, so sequences do not match perfectly
- Computationally efficient: some algorithms are designed for use in databases involving millions of sequences
- Provide diagnostic tools of dealing with alignment issues: many unknown sequences do not have a clear start and finish
- Assume that irrelevant information is embedded in the sequence, analogous to noise in event sequences
- Existing sites are generally designed to search against a very large base of known DNA and protein sequences, which we now have with GDELT

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

Technical models and elite human forecasters, developed by multiple research groups on a wide variety of indicators, can forecast a variety of indicators of political conflict at 6 to 24 month horizons at around 80% out-of-sample accuracy.

This is 20% to 30% more accurate than typical human forecasting.

Major lessons learned so far

Technical models and elite human forecasters, developed by multiple research groups on a wide variety of indicators, can forecast a variety of indicators of political conflict at 6 to 24 month horizons at around 80% out-of-sample accuracy.

This is 20% to 30% more accurate than typical human forecasting.

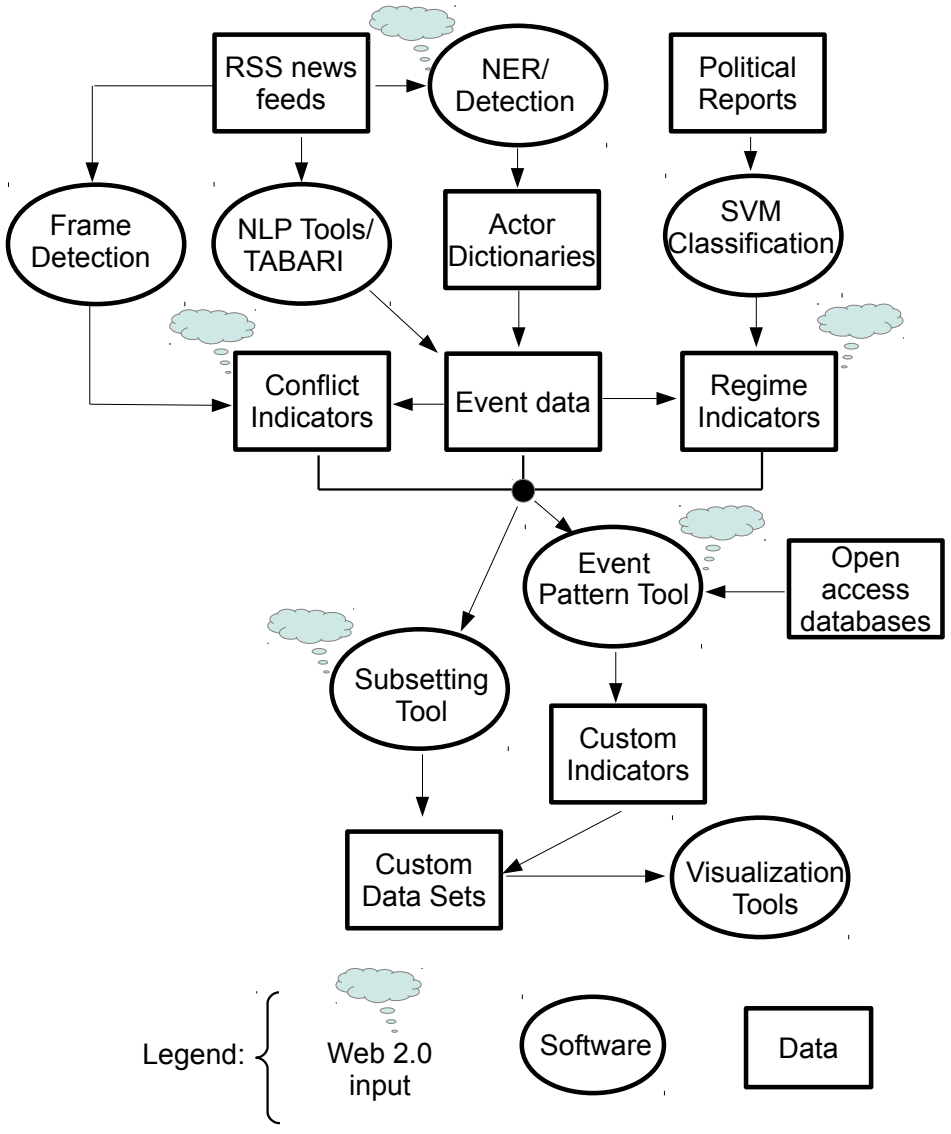
Major lessons learned so far

- 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
- Real-time forecasting models are undeveloped, though this is likely to change with the availability of GDELT and other real-time data sets (e.g. new social media)
- At what time horizon and with what pattern do errors occur in high density/long time-series datasets

Multi-Attribute Data Collected on Web (MADCOW)



Thank you

Email: schrodt735@gmail.com

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

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