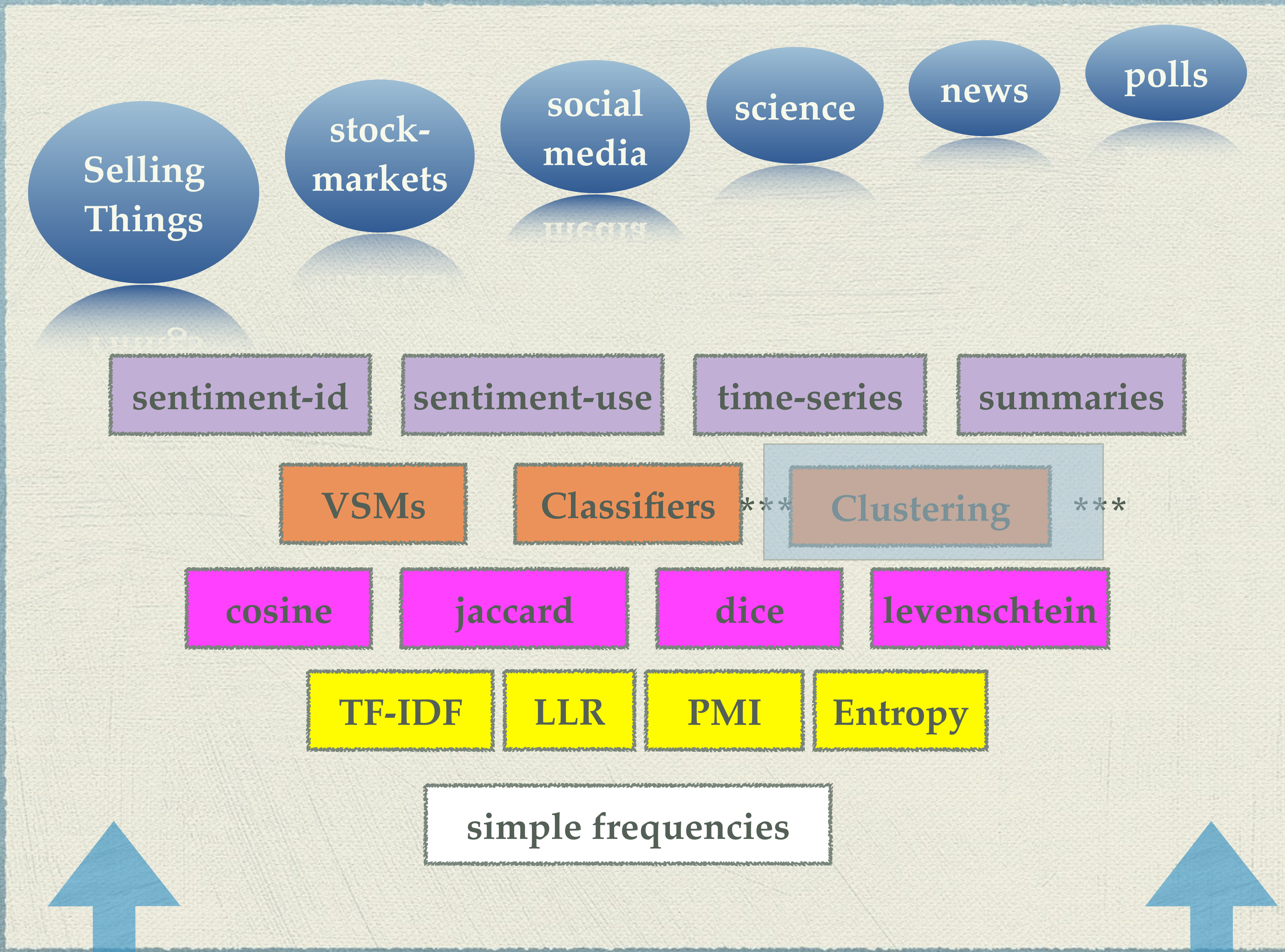
Clustering

Lecture 6: Text Analytics for Big Data

Mark Keane, Insight/CSI, UCD


Selling Things
stock-markets
social media
science
news
polls
sentiment-id
sentiment-use
time-series
summaries
VSMs
Classifiers
***
Clustering
***
cosine
jaccard
dice
levenschtein
TF-IDF
LLR
PMI
Entropy
simple frequencies

Outline

- Why Do This?
- Sometimes you have no labels, but rather want to *discover* the categories…
- Clustering (with Unsupervised Techniques)
- Tangent on Evaluation: Validating Clusters

Unsupervised: Clustering

- K-Means
- Hierarchical Clustering
- Graph-Based Clustering

Clustering

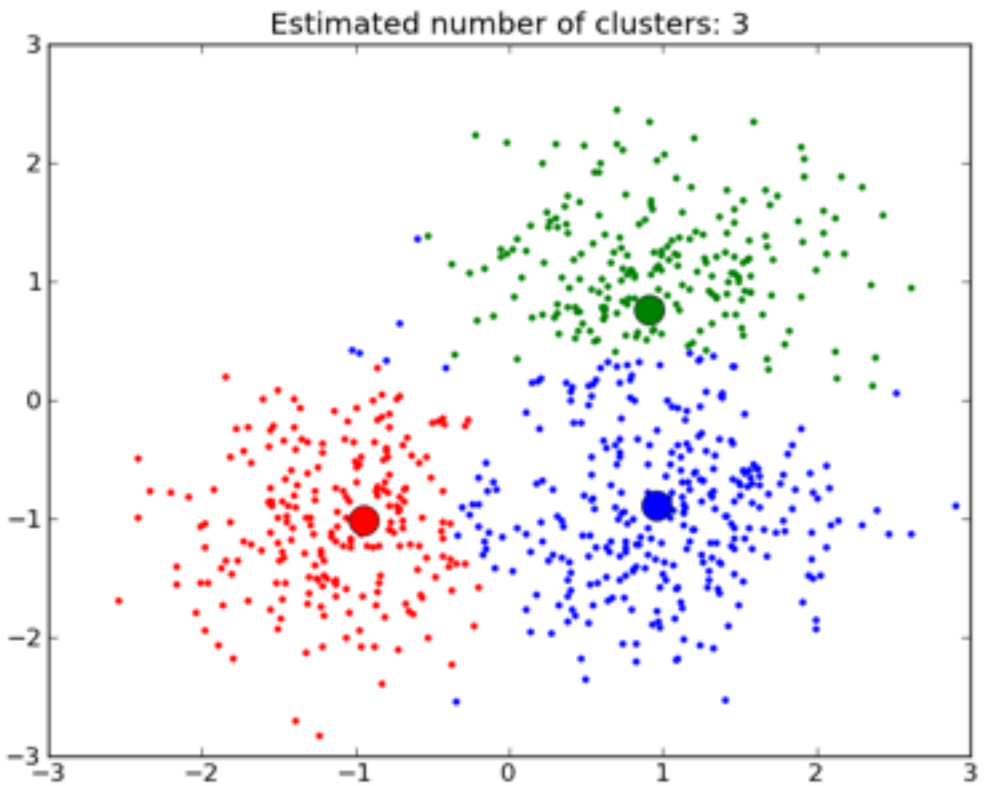
Why We Do This?

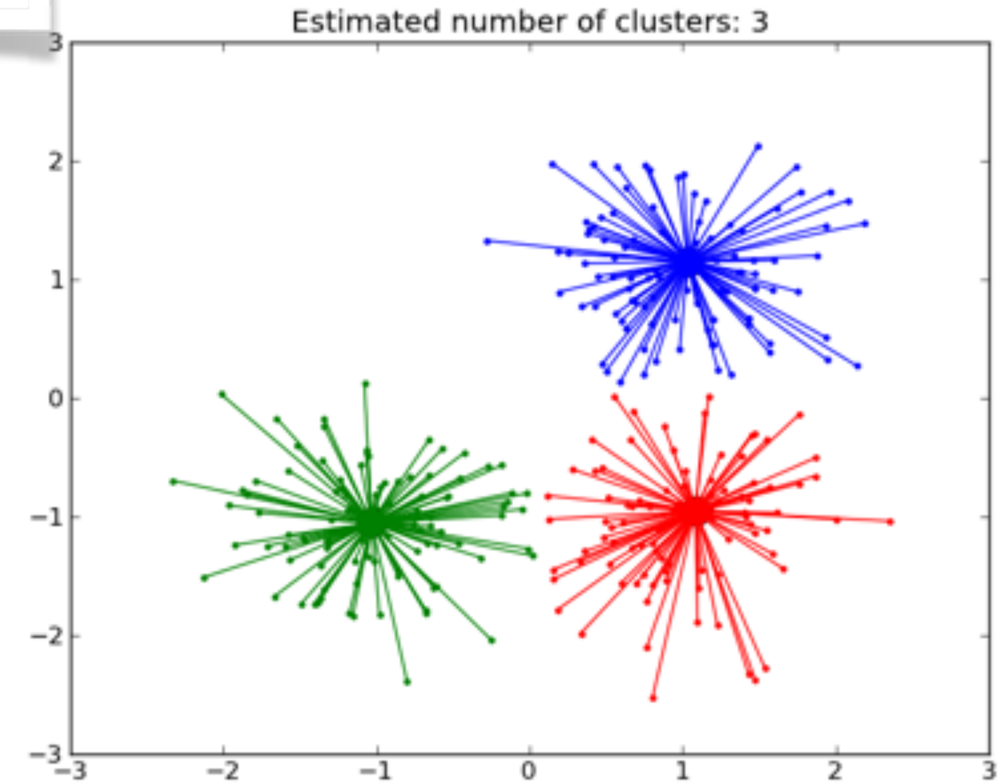
To Cluster…

- Clustering is used in text analytics to discover groups of items, to circumscribe…
- Do these text-items form a “natural” group that we find to be spam or non-spam?
- If these items have these features they group together; like birds of a feather…

Why Clustering?

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

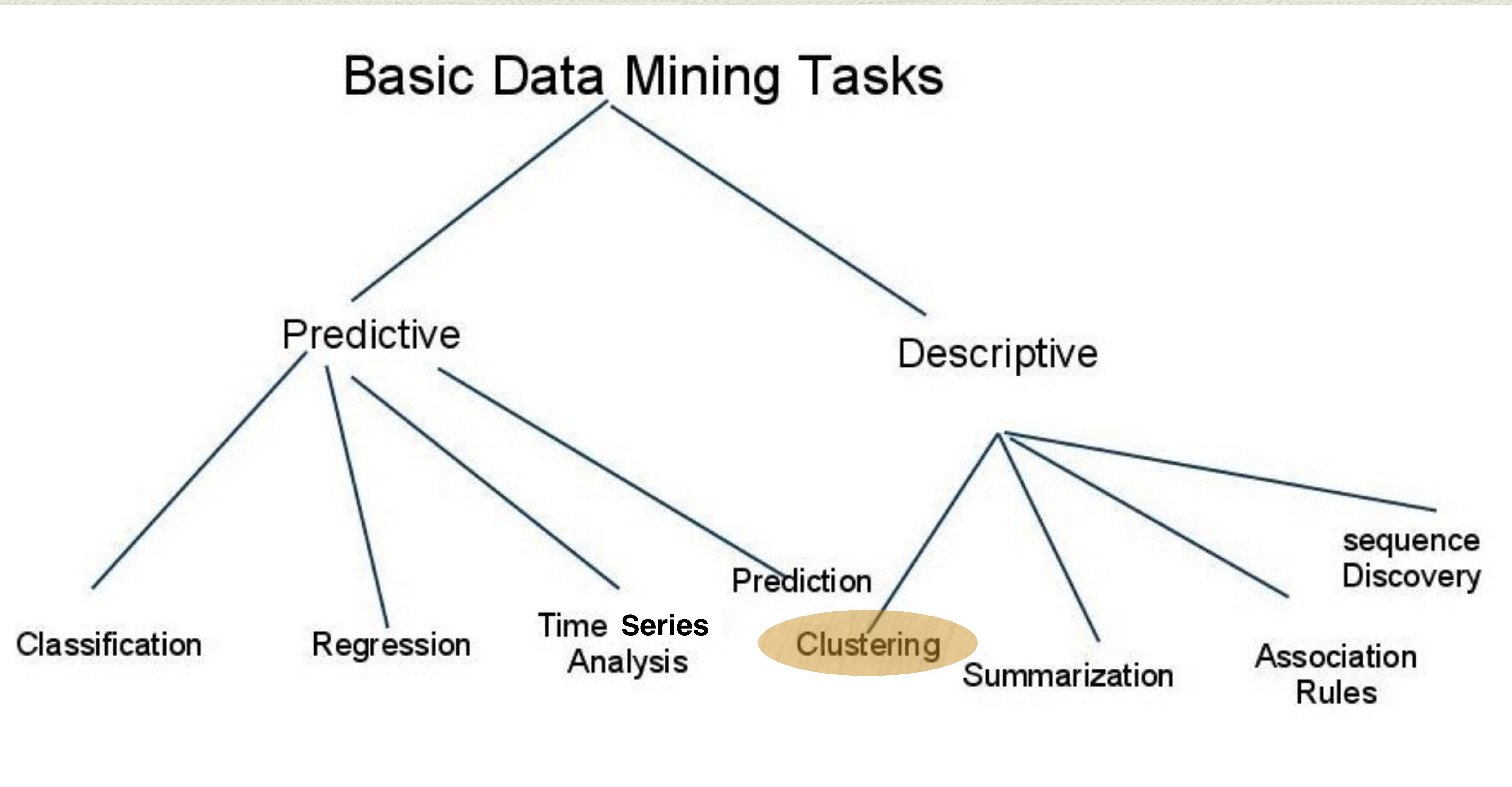




To Cluster…

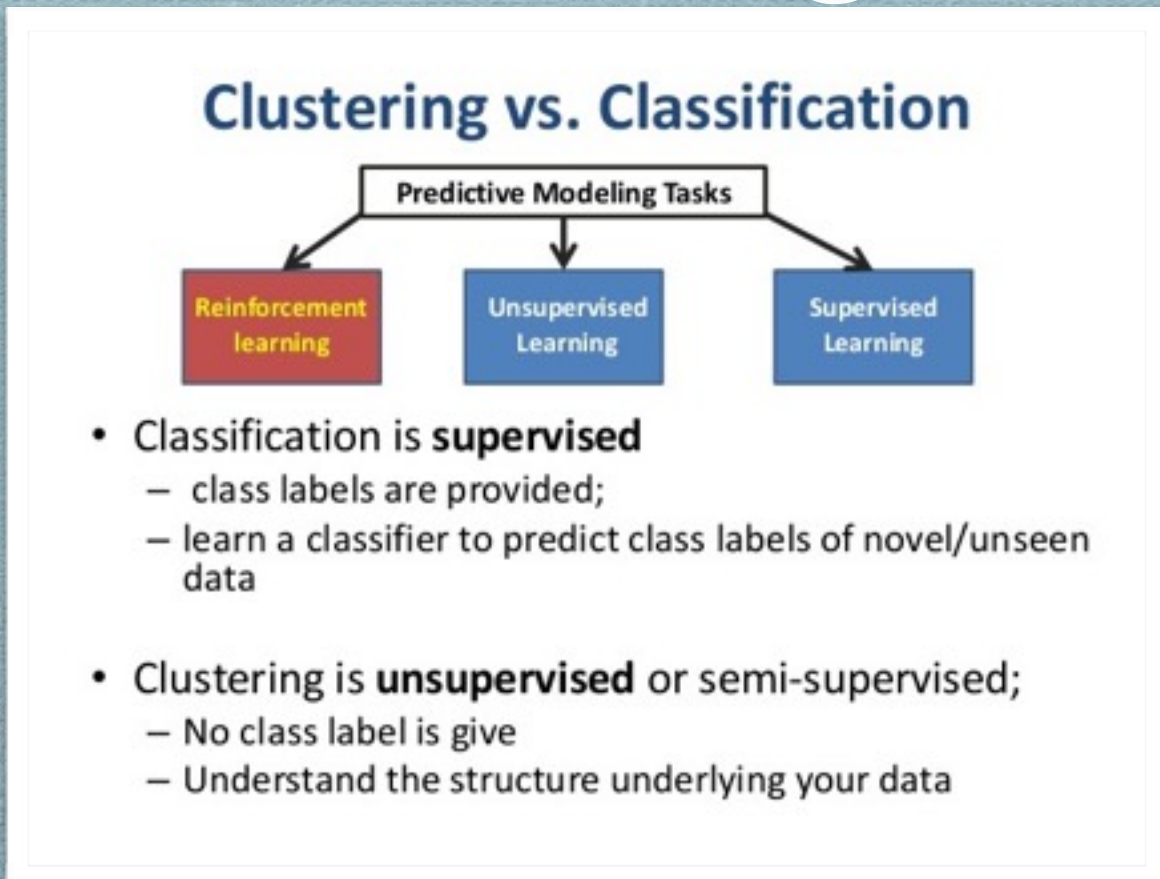
- Clustering is the unsupervised face of machine learning; when data is unlabelled
- It gives you a description of possible classes, based on how they group together
- As such, it may be a preliminary to classification; but it is a task in itself

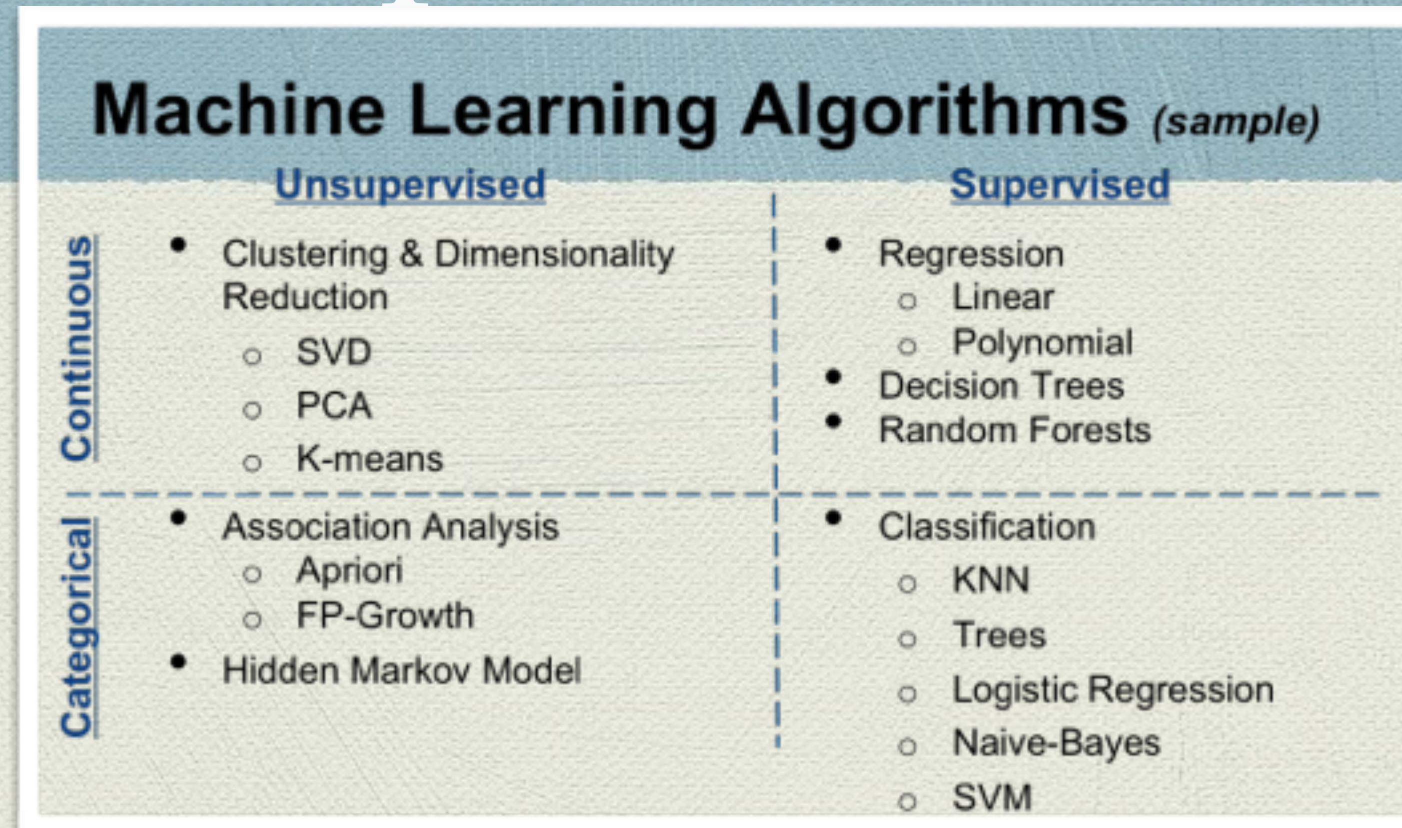
One View of Tasks…

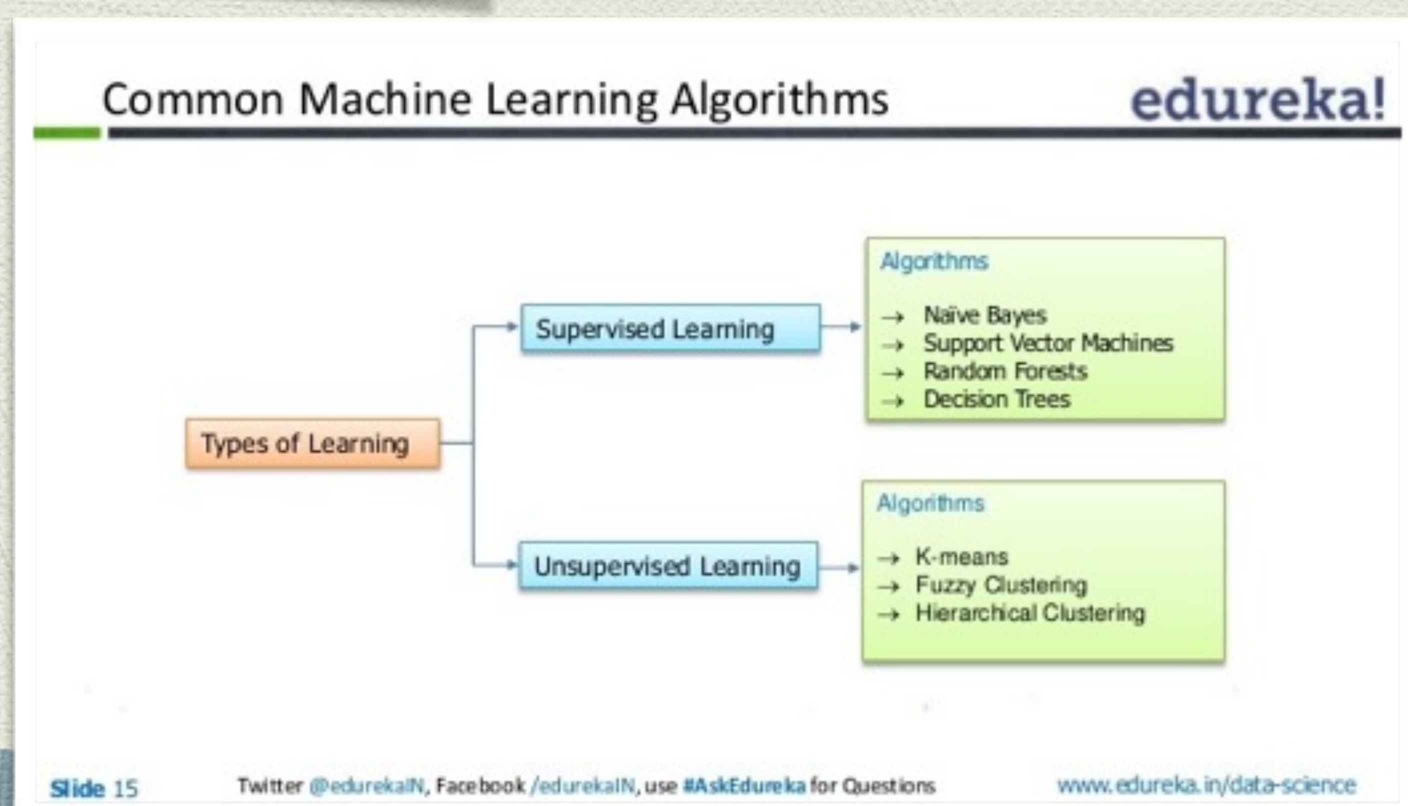


http://www.slideshare.net/KapilRode/data-mining-32694954

Clustering is Unsupervised…



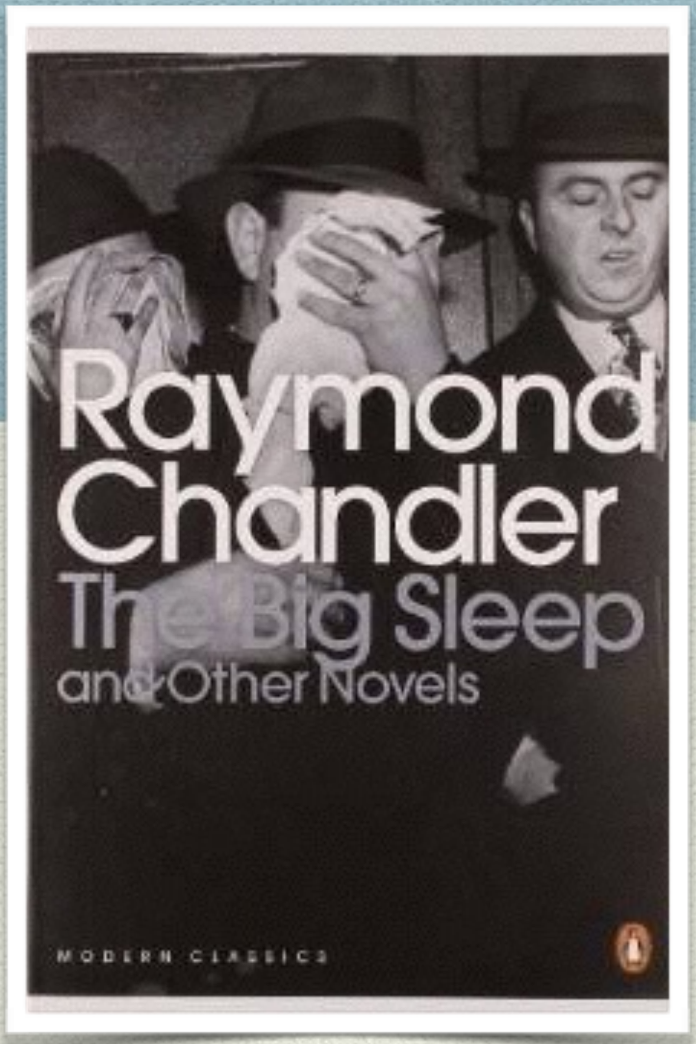


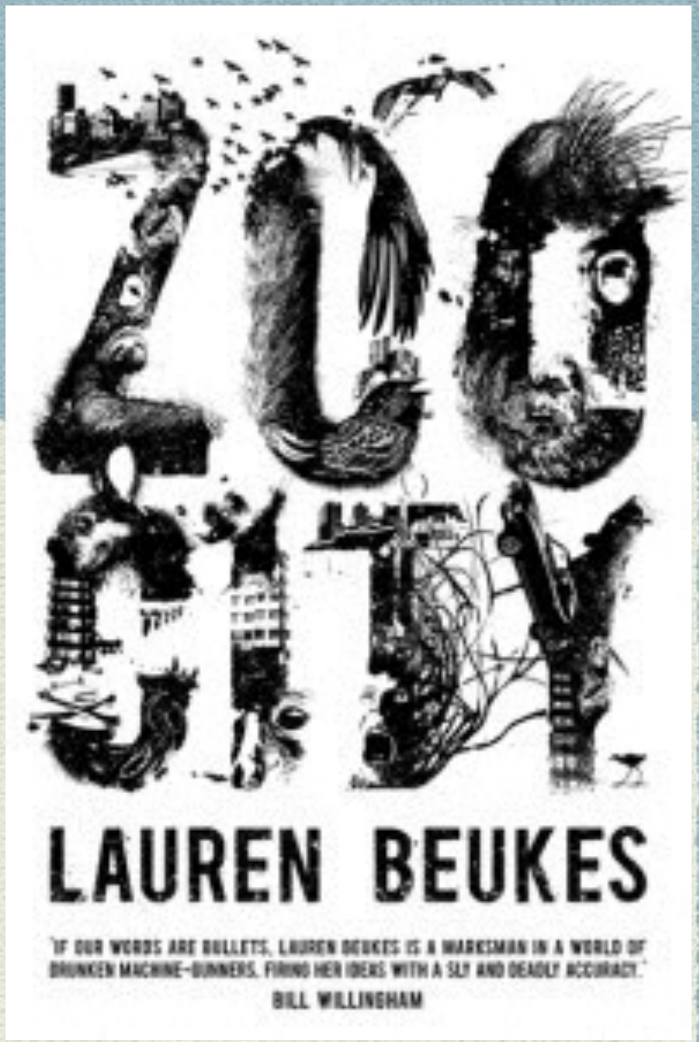


Clustering is Ubiquitous

- We have seen similarity, as a way to compare individual items to one another (shopper1 is like shopper 2)
- Sometimes you want to find a group of things; so you can say *x1*,*x2*,*x3* are from Group-*X*; *shopper1*, *shopper2*, *shopper3* are *BingeShoppers*, but *shopper4*, *shopper5*, *shopper6* are *YoungFamilyShoppers*
- X is a general class of things, a group, as is BingeShopper
- Many ML techniques use different **clustering techniques**; there are a wide variety of options

Examples





- Activity profiles of Tweeters suggest 3 groups: lurkers, actives and celebs
- Company reports divide into negative or positive groups
- Urban-fantasy-crime-readers are a different group than traditional noir-crime-readers

Selecting Some…

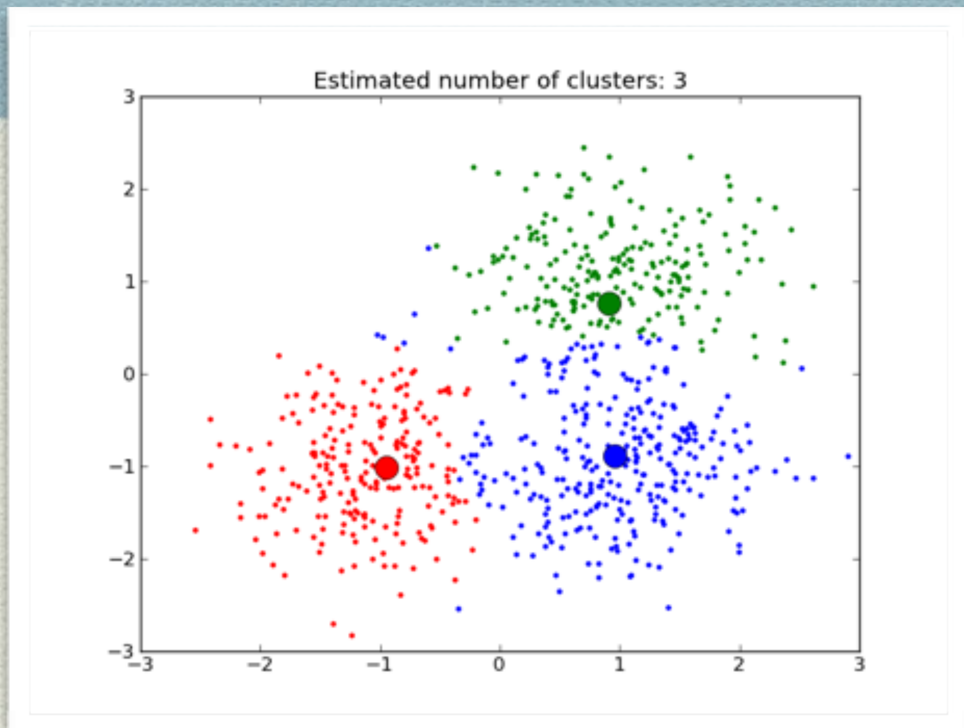
- **Supervised Learning: Classification (last lecture)**
  - K Nearest Neighbours
  - Naive Bayes
  - Logistic Regression
  - Support Vector Machine
- **Unsupervised Learning: Clustering**
  - K-Means
  - Hierarchical Clustering
  - Graph-Based Clustering

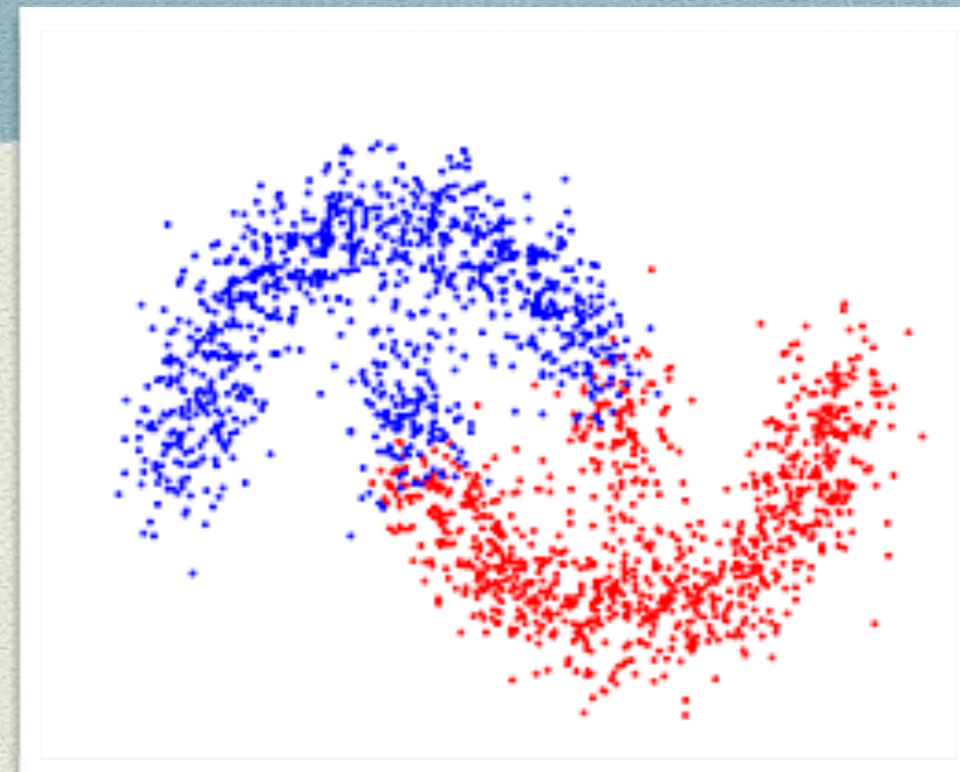
http://www.slideshare.net/mirjalil/clustering-on-database-systems-rkm-42588852

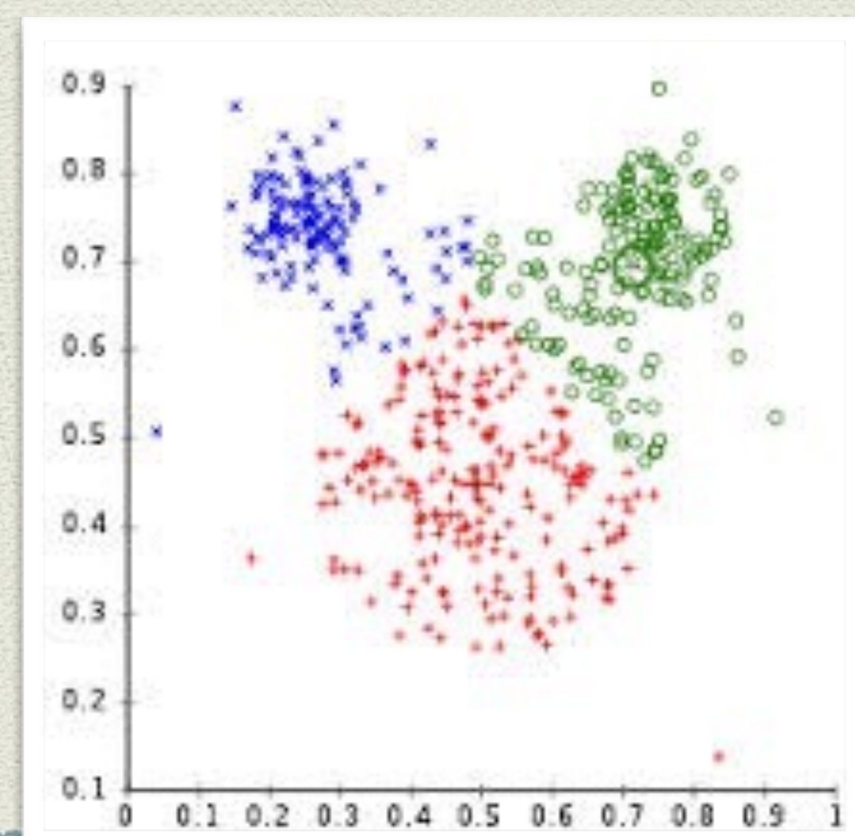
Clustering

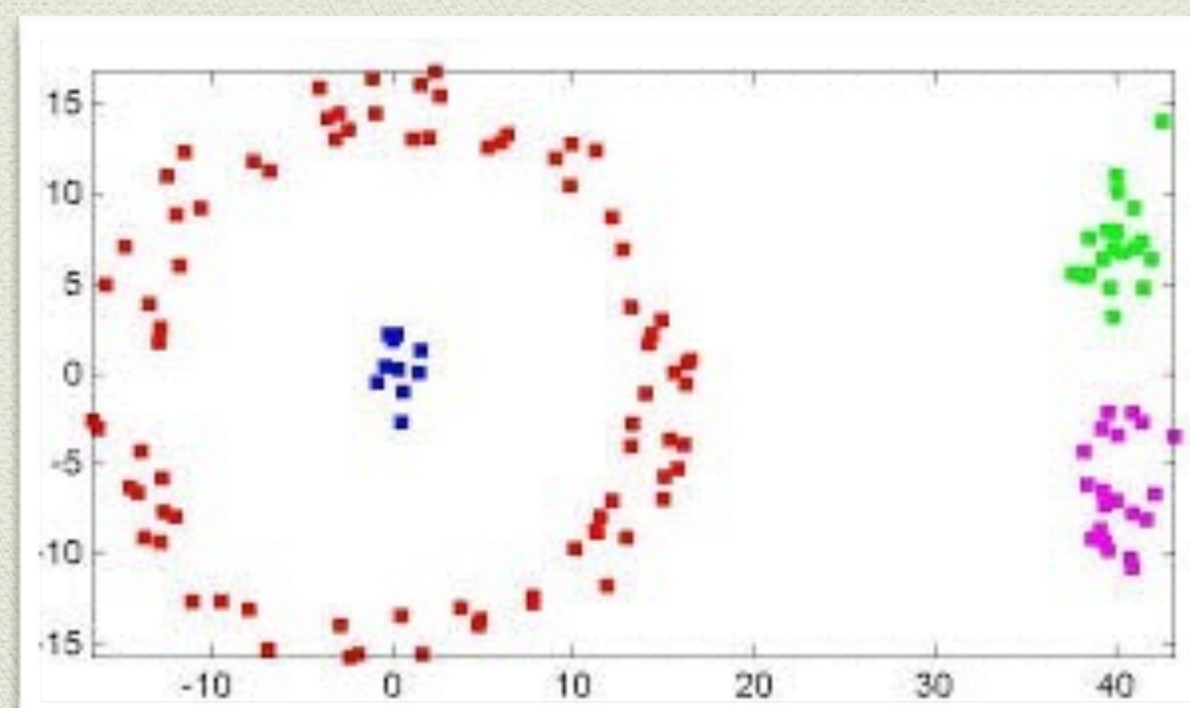
Unsupervised Techniques

Clustering: Pics









Unsupervised Techniques

Unsupervised learning

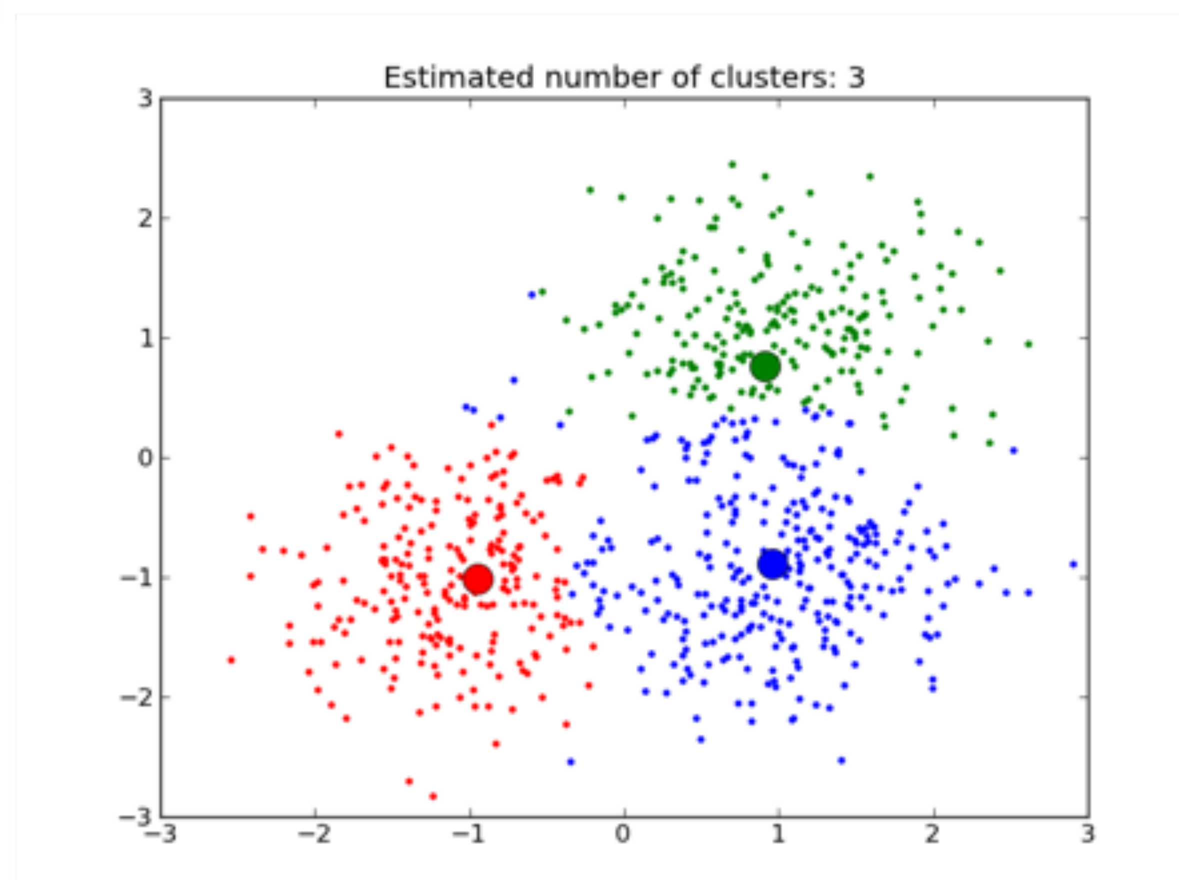
From Wikipedia, the free encyclopedia

In machine learning, the problem of **unsupervised learning** is that of trying to find hidden structure in unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning and reinforcement learning.

Unsupervised learning is closely related to the problem of density estimation in statistics.[1] However unsupervised learning also encompasses many other techniques that seek to summarize and explain key features of the data. Many methods employed in unsupervised learning are based on data mining methods used to preprocess[*citation needed*] data.

Approaches to unsupervised learning include:

- clustering (e.g., k-means, mixture models, hierarchical clustering),[2]
- Approaches for learning latent variable models such as
  - Expectation–maximization algorithm (EM)
  - Method of moments
  - Blind signal separation techniques, e.g.,
    - Principal component analysis,
    - Independent component analysis,
    - Non-negative matrix factorization,
    - Singular value decomposition.[3]



Start from a set of items and analyse them to find clusters

Supervised: Clustering

- K-Means
- Hierarchical Clustering
- Graph-Based Clustering

Clustering

k-means

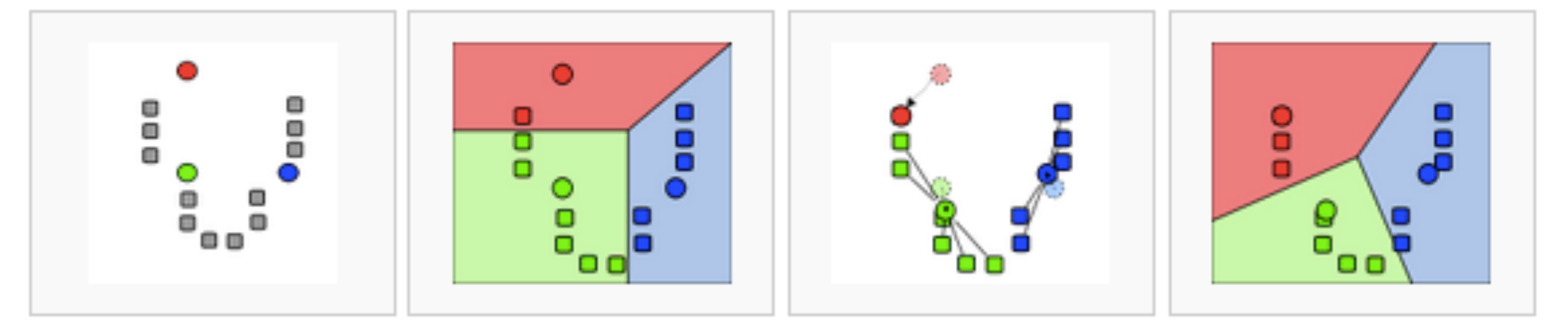
- Take set of animals described by their features and then group them on these features

Wookies

Pandas

k-means: Idea

- Randomly pick some points in m-d space
- Get distance of all instances from these points, assign to group if closest to point
- Move the points to the *centre* of these groups
- Repeat until you're no longer moving points



en.wikipedia.org/wiki/K-means_clustering

k-means: Formula

Given a set of observations ($\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k ($\leq n$) sets $\mathbf{S} = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares (WCSS). In other words, its objective is to find:

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \left\| \mathbf{x} - \boldsymbol{\mu}_i \right\|^2$$

where $\boldsymbol{\mu}_i$ is the mean of points in S_i.

Algorithm 1 The k-means algorithm

Input: set D, distance measure $dist$, number k of cluster

Output: A partitioning $\mathbb{P}$ of the set D of documents (i.e., a set $\mathbb{P}$ of k disjoint subsets of D with $\bigcup_{P \in \mathbb{P}} P = D$).

```
Choose randomly k data points from D as starting centroids t_P1 ... t_Pk.
repeat
    Assign each point of P to the closest centroid with respect to dist.
    (Re-)calculate the cluster centroids t_P1 ... t_Pk of clusters P1 ... Pk.
until cluster centroids t_P1 ... t_Pk are stable
return set ℙ := {P1, ..., Pk}, of clusters.
```

Good

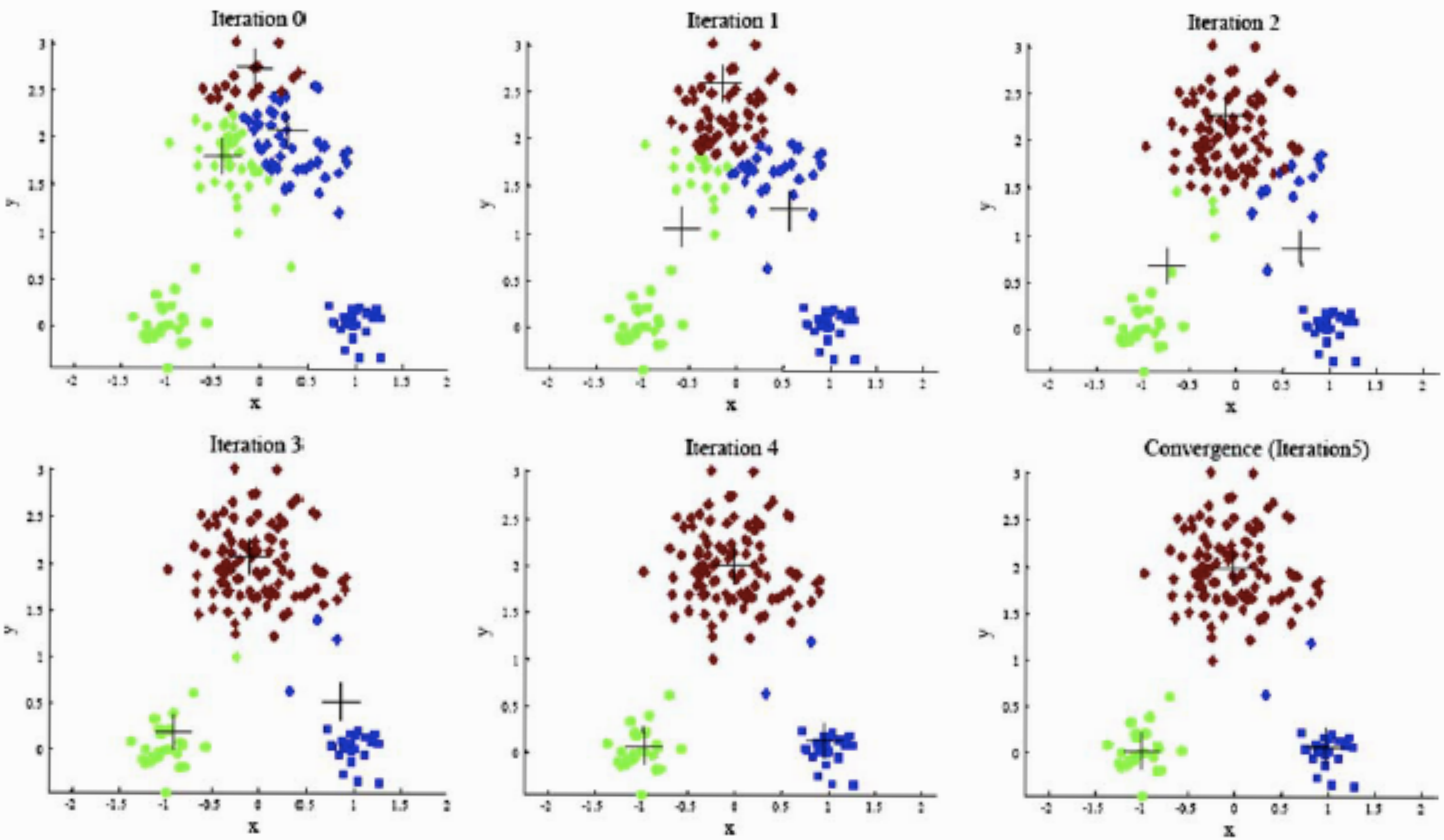


Fig. 1 Changes in cluster representative locations (indicated by '+' signs) and data assignments (indicated by color) during an execution of the k-means algorithm

Wu, X., et al. (2008). Top 10 algorithms in data mining. Knowledge and Information Systems, 14(1), 1-37.

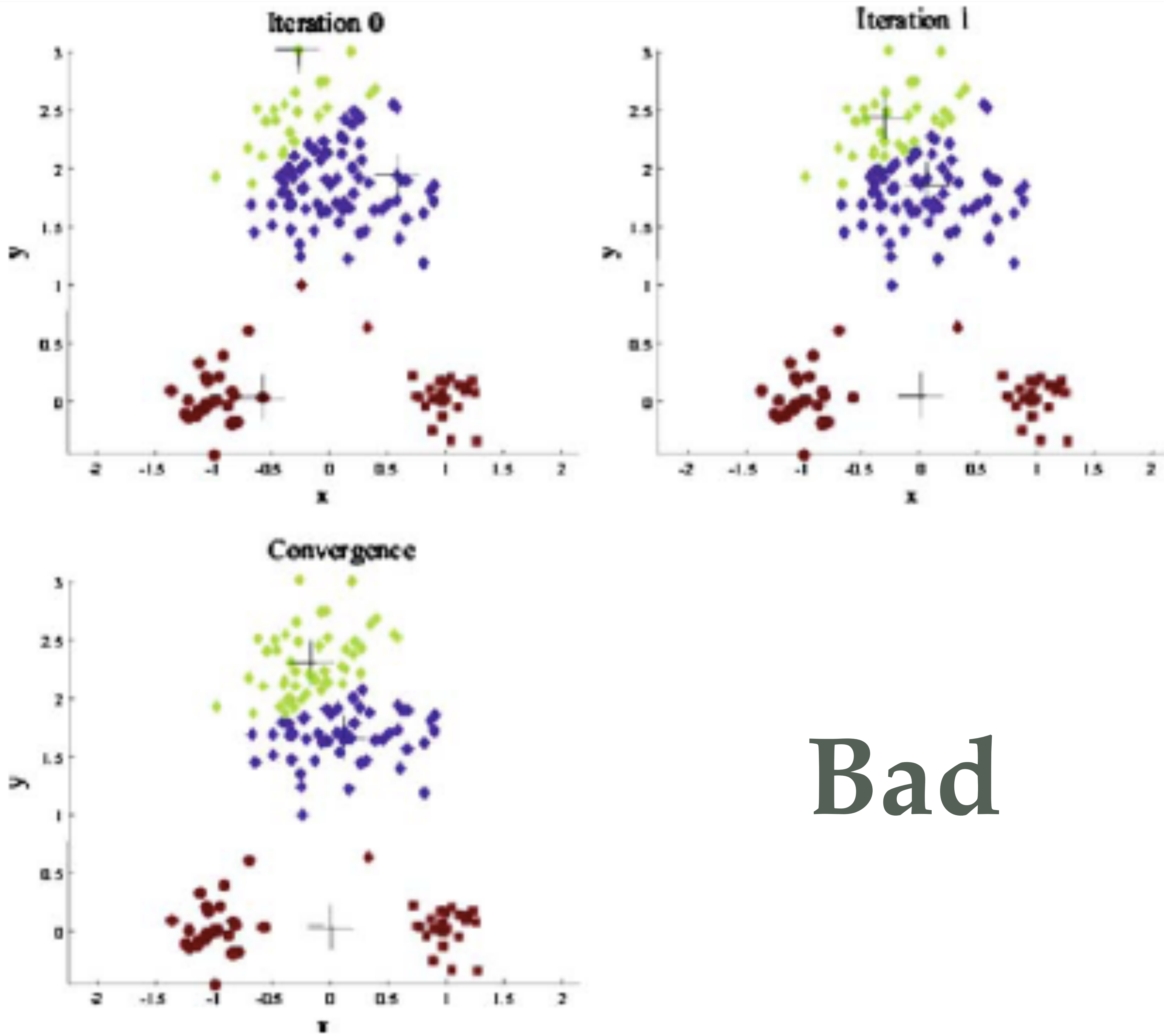


Fig. 2 Effect of an inferior initialization on the k-means results

Bad

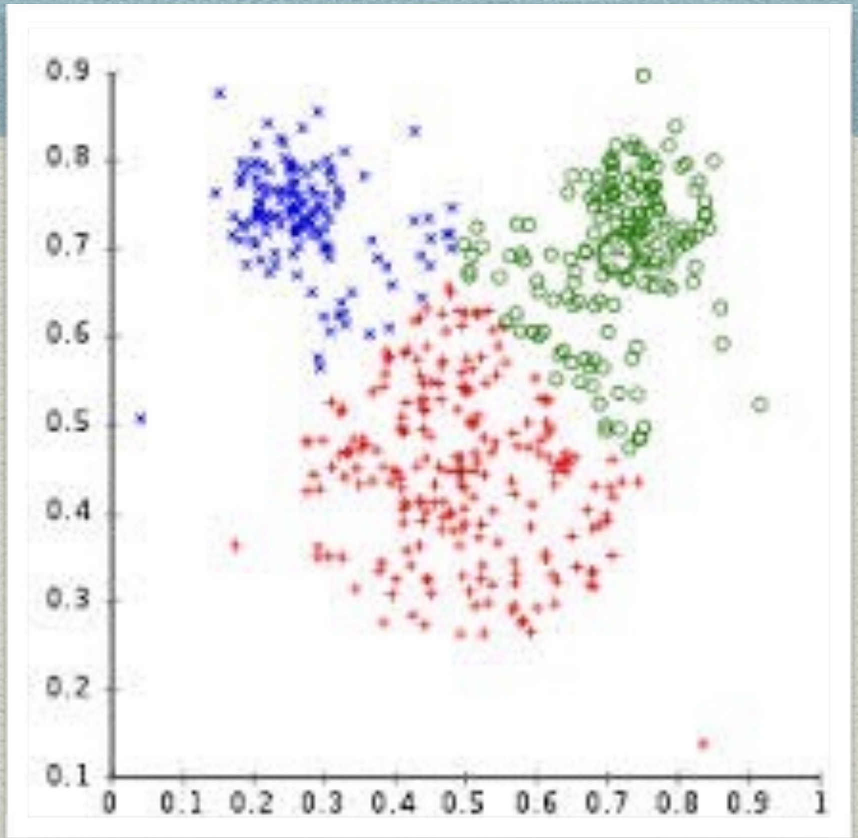
Wu, X., et al. (2008). Top 10 algorithms in data mining. Knowledge and Information Systems, 14(1), 1-37.

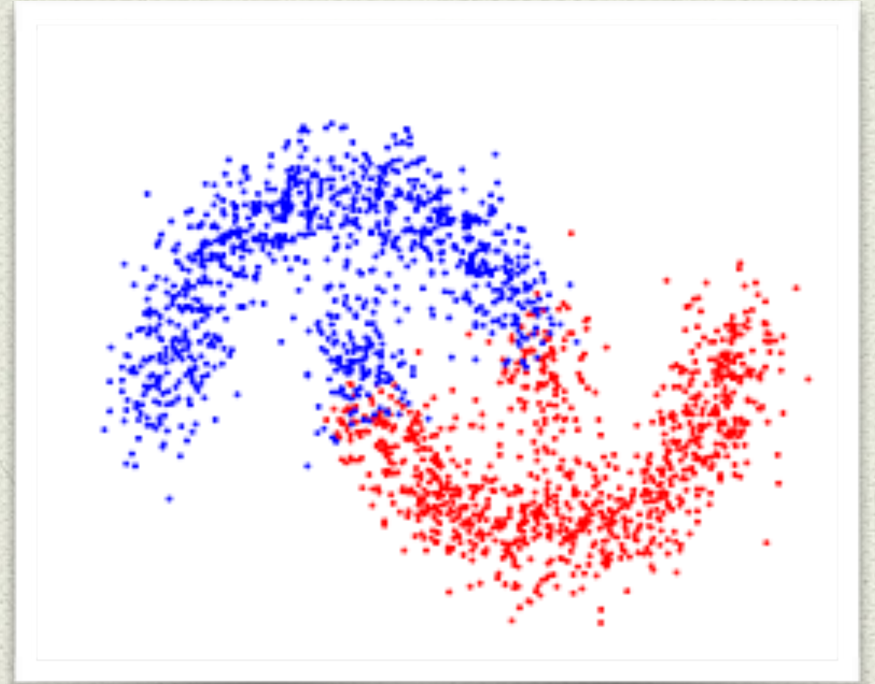
k-means: Issues I

- May not find global optimum (random start points); though there are some heuristics to improve initial point choice
- May not find global optimum (no. of points/groups), but *diagnostic* methods find no. of groups and cost fn methods exist
- NP-hard Euclidean space, even for 2 clusters; but heuristics help
- Can use other distance-metrics but compromise convergence
- In practice, often do 1000 iterations of 2-points, 3-points, etc; and find the most persistent group that emerges running all night

k-means: Issues II

- Data really needs to be reasonably separated, spherical clusters
- Poor if there are non-convex shaped clusters in the data





k-means Eg: Text Classification

- Classic use is in text processing; recall, in VSMs, docs are vectors, so each is a point in an m-d space
- Euclidean distance is the metric between these points, then you can see how k-means can be used
- Use entropy and F measure can be used to determine goodness of clusters; entropy why?

Steinbach, M., Karypis, G., & Kumar, V. (2000, August). A comparison of document clustering techniques. In KDD workshop on text mining (Vol. 400, No. 1, pp. 525-526).

k-means Eg: Text Classification

Table 1: Summary description of document sets.

Data Set	Source	Documents	Classes	Words
re0	Reuters	1504	13	11465
re1	Reuters	1657	25	3758
wap	WebAce	1560	20	8460
tr31	TREC	927	7	10128
tr45	TREC	690	10	8261
fbis	TREC	2463	17	
la1	TREC	3204	6	
la2	TREC	3075	6	

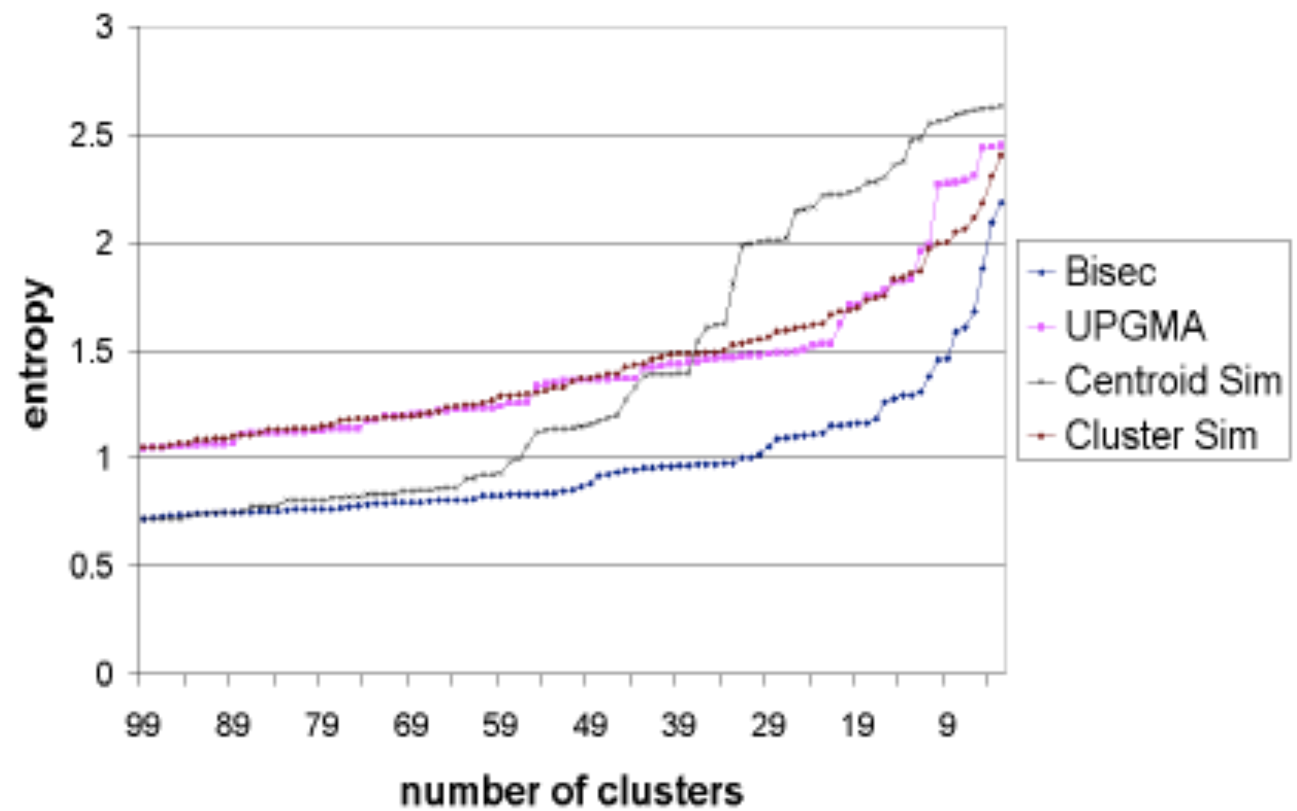


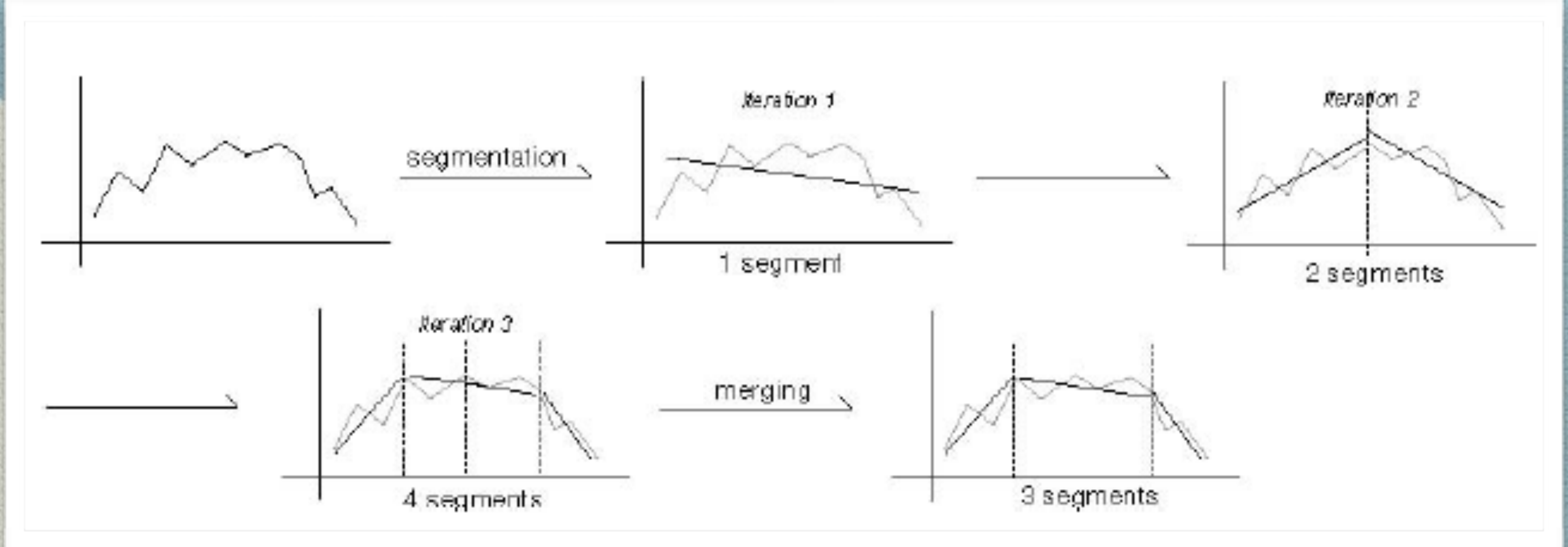
Figure 1: Comparison of entropy for re0 data set.

as Cluster-No decreases
Entropy increases, why?

Steinbach, M., Karypis, G., & Kumar, V. (2000, August). A comparison of document clustering techniques. In KDD workshop on text mining (Vol. 400, No. 1, pp. 525-526).

k-means Eg: Stocks & News

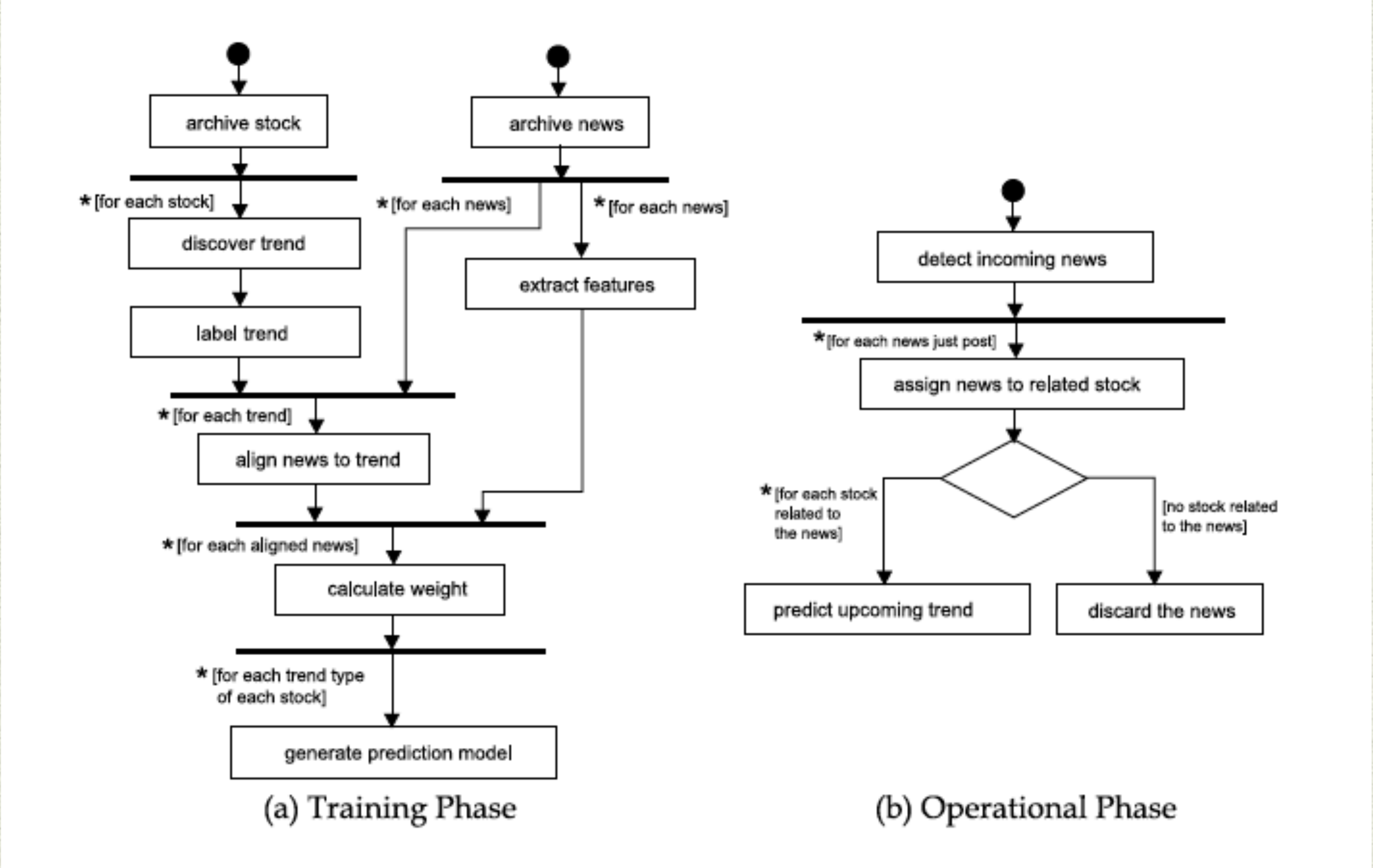
Traced rising/ falling trends in stock prices



- Gathered articles within rising or falling period and clustered using k-means
- Compute similarities 'tween news-article group to to relate group to rise/fall trend

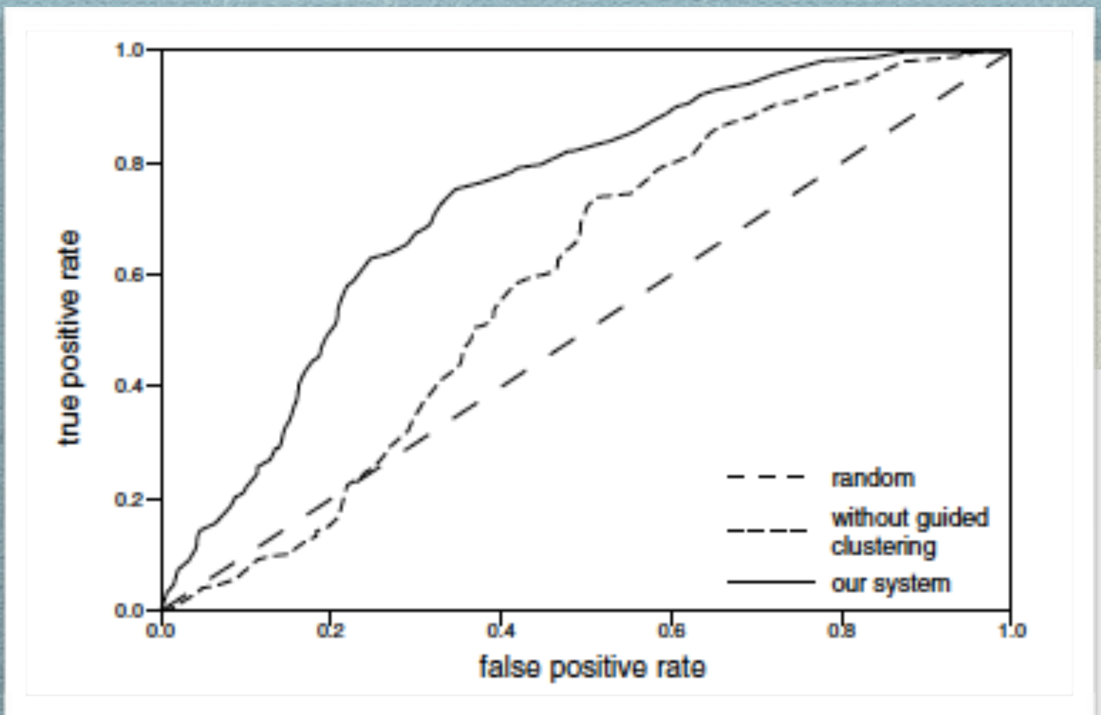
Fung, G. P. C., Yu, J. X., & Lam, W. (2002). News sensitive stock trend prediction. In *Advances in knowledge discovery and data mining* (pp. 481-493). Springer Berlin Heidelberg.

k-means #2: Stock Prediction



Fung, G. P. C., Yu, J. X., & Lam, W. (2002). News sensitive stock trend prediction. In *Advances in knowledge discovery and data mining* (pp. 481-493). Springer Berlin Heidelberg.

k-means #2: Stock Prediction



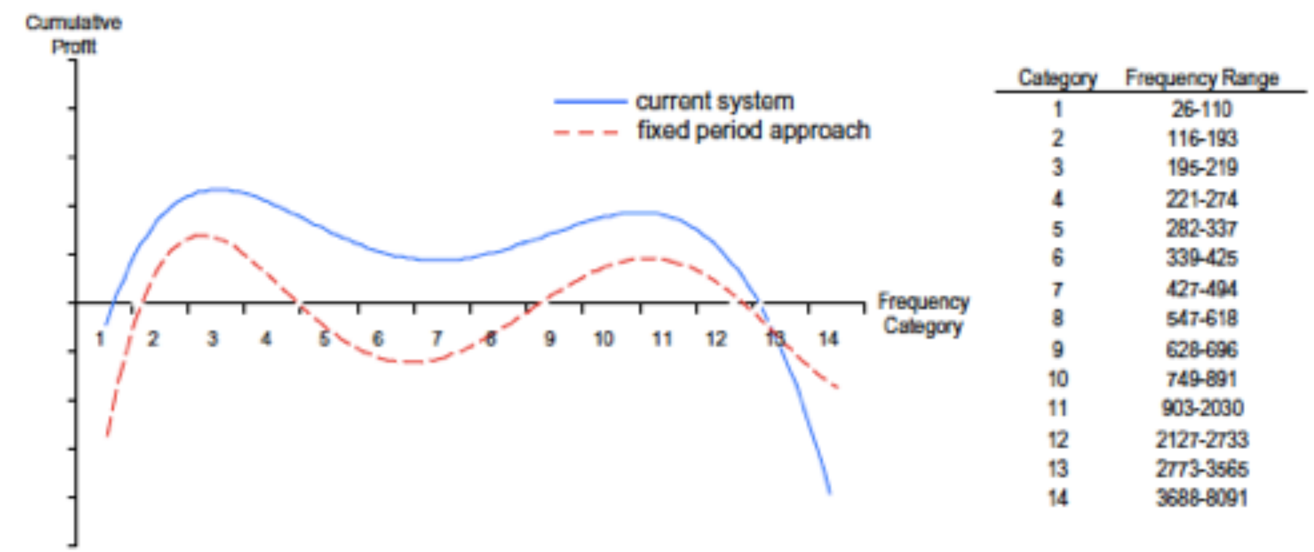


Fig. 6. The relationship between the frequency of news announced and the resulting profit

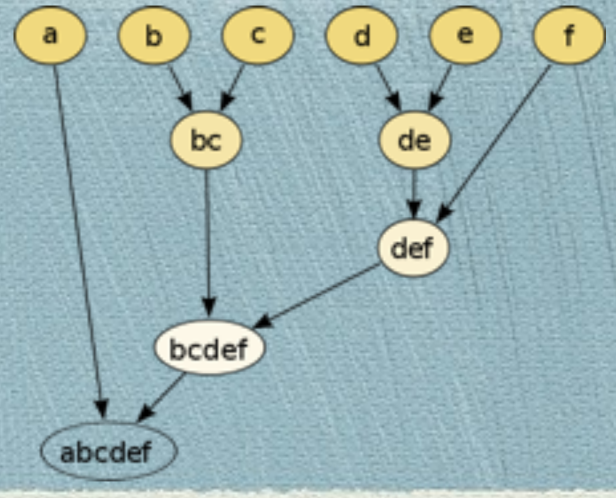
Fung, G. P. C., Yu, J. X., & Lam, W. (2002). News sensitive stock trend prediction. In *Advances in knowledge discovery and data mining* (pp. 481-493). Springer Berlin Heidelberg.

Clustering

Hierarchical Clustering

(HCA-Hierarchical Cluster Analysis)

HCA: The Idea



- Start with a set of items and a distance metric
- Find the (similarity) distance ‘tween all items
- Merge the two shortest-distance items into a group; then get distance ‘tween this group and other items
- Merge the items (group/singles) that have the shortest distance on this iteration
- Keep iterating …

http://en.wikipedia.org/wiki/Hierarchical_clustering

HCA: Types

- *Agglomerative HCA*: where you start bottom-up from individual items (eg docs) and merge them on similarity (shortest distance)
- *Divisive HCA**: where you start top-down from one cluster and sub-divide it by difference (longest distance)

* in practice, works poorly

C. D. Manning & H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, Camb., MA, 2001.

HCA: Formula

A frequently used distance measure is the Euclidian distance. We calculate the distance between two text documents $d_1, d_2 \in D$ as follows:

$$dist(d_1, d_2) = \sqrt[2]{\sum_{k=1}^{m} |w(d_1, t_k) - w(d_2, t_k)|^2} \ . \qquad (4)$$

However, the Euclidean distance should only be used for normalized vectors, since otherwise the different lengths of documents can result in a smaller distance between documents that share less words than between documents that have more words in common and should be considered therefore as more similar.

Algorithmic steps for Agglomerative Hierarchical clustering

Let $X = \{x_1, x_2, x_3, ..., x_n\}$ be the set of data points.

1) Begin with the disjoint clustering having level L(0) = 0 and sequence number m = 0.

2) Find the least distance pair of clusters in the current clustering, say pair (r), (s), according to d[(r),(s)] = min d[(i),(j)] where the minimum is over all pairs of clusters in the current clustering.

3) Increment the sequence number: m = m +1.Merge clusters (r) and (s) into a single cluster to form the next clustering m. Set the level of this clustering to L(m) = d[(r),(s)].

4) Update the distance matrix, D, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The distance between the new cluster, denoted (r,s) and old cluster(k) is defined in this way: d[(k), (r,s)] = min (d[(k),(r)], d[(k),(s)]).

5) If all the data points are in one cluster then stop, else repeat from step 2).

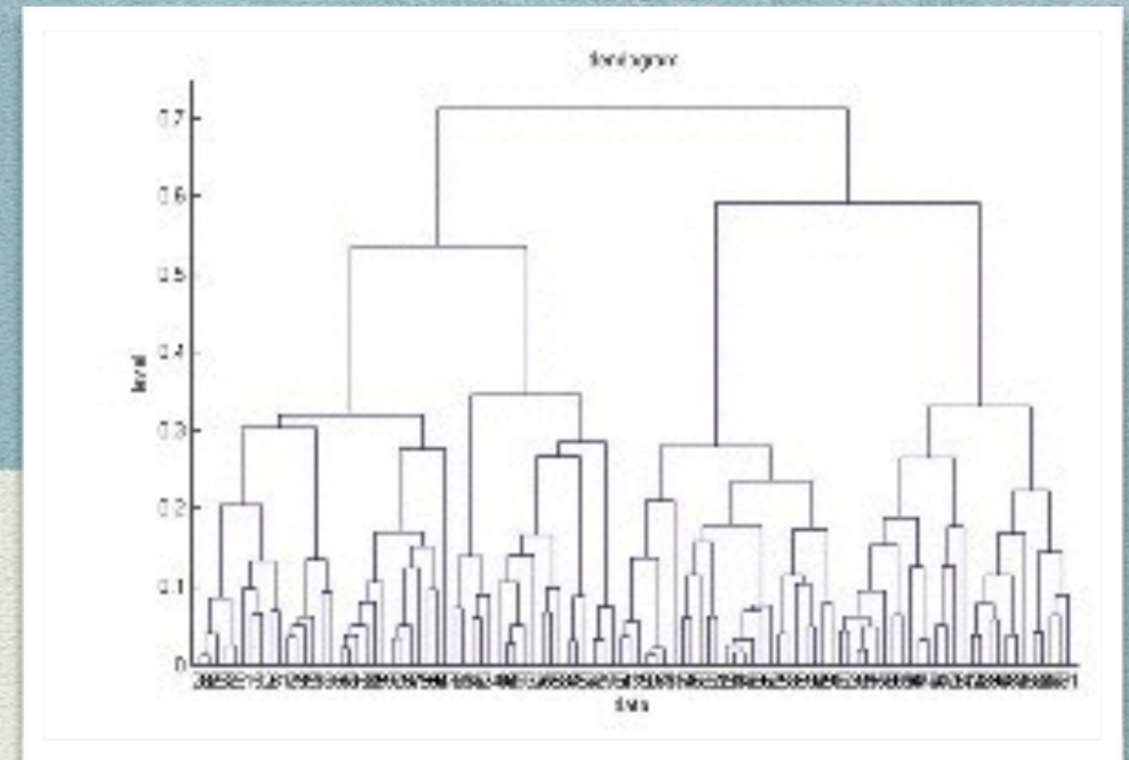
Divisive Hierarchical clustering - It is just the reverse of Agglomerative Hierarchical approach.

http://sites.google.com/site/dataclusteringalgorithms/hierarchical-clustering-algorithm

HCA: One Issue

- *How to you compare a group to other item/group?:*
  - use the centroid of the group
  - use the average of the group
  - use the max/min of distance between items in the group
- More complicated methods:
  - Decrease in variance of group being merged (*Ward's Criterion*)
  - Probability that groups come from same distribution

HCA: Pros/Cons



PROS:

- works well most of the time
- a priori number of clusters not required

CONS:

- Correct no of clusters can be hard to identify
- Depending on *dist*: (i) sensitive to noise and outliers, (ii) breaking large clusters, (ii) handling different-sized clusters and convex shapes
- Complexity in memory and time

```python
import numpy as np
import scipy.cluster.hierarchy as hac
import matplotlib.pyplot as plt


a = np.array([[0.1,   2.5],
              [1.5,   .4 ],
              [0.3,   1  ],
              [1  ,   .8 ],
              [0.5,   0  ],
              [0  ,   0.5],
              [0.5,   0.5],
              [2.7,   2  ],
              [2.2,   3.1],
              [3  ,   2  ],
              [3.2,   1.3]])

fig, axes23 = plt.subplots(2, 3)

for method, axes in zip(['single', 'complete'], axes23):
    z = hac.linkage(a, method=method)

    # Plotting
    axes[0].plot(range(1, len(z)+1), z[::-1, 2])
    knee = np.diff(z[::-1, 2], 2)
    axes[0].plot(range(2, len(z)), knee)

    num_clust1 = knee.argmax() + 2
    knee[knee.argmax()] = 0
    num_clust2 = knee.argmax() + 2

    axes[0].text(num_clust1, z[::-1, 2][num_clust1-1], 'possible\n<- knee point')

    part1 = hac.fcluster(z, num_clust1, 'maxclust')
    part2 = hac.fcluster(z, num_clust2, 'maxclust')

    clr = ['#2200CC' ,'#D9007E' ,'#FF6600' ,'#FFCC00' ,'#ACE600' ,'#0099CC' ,
    '#8900CC' ,'#FF0000' ,'#FF9900' ,'#FFFF00' ,'#00CC01' ,'#0055CC']
```

http://stackoverflow.com/questions/21638130/tutorial-for-scipy-cluster-hierarchy

HCA Eg: Text Clustering

- We saw earlier; dividing a set of documents into a number of classes

Table 1: Summary description of document sets.

Data Set	Source	Documents	Classes	Words
re0	Reuters	1504	13	11465
re1	Reuters	1657	25	3758
wap	WebAce	1560	20	8460
tr31	TREC	927	7	10128
tr45	TREC	690	10	8261
fbis	TREC	2463	17	2000
la1	TREC	3204	6	31472
la2	TREC	3075	6	31472

Steinbach, M., Karypis, G., & Kumar, V. (2000, August). A comparison of document clustering techniques. In KDD workshop on text mining (Vol. 400, No. 1, pp. 525-526).

HCA Eg: Clustering Methods

4. Agglomerative Hierarchical Techniques

We used three different agglomerative hierarchical techniques for clustering documents.

Intra-Cluster Similarity Technique: This hierarchical technique looks at the similarity of all the documents in a cluster to their cluster centroid and is defined by Sim(X) = $\sum_{d \in X} cosine(\mathbf{d}, \mathbf{c})$, where $\mathbf{d}$ is a document in cluster, X, and $\mathbf{c}$ is the centroid of cluster X, i.e., the mean of the document vectors. The choice of which pair of clusters to merge is made by determining which pair of clusters will lead to smallest decrease in similarity.

Centroid Similarity Technique: This hierarchical technique defines the similarity of two clusters to be the cosine similarity between the centroids of the two clusters.

UPGMA: This is the UPGMA scheme as described in [2]. It defines the cluster similarity as follows, where d_1 and d_2 are documents in cluster1 and cluster2, respectively.

$$similarity(cluster1, cluster2) = \frac{\sum cosine(\mathbf{d}_1, \mathbf{d}_2)}{size(cluster1) * size(cluster2)}$$

Steinbach, M., Karypis, G., & Kumar, V. (2000, August). A comparison of document clustering techniques. In KDD workshop on text mining (Vol. 400, No. 1, pp. 525-526).

HCA Eg: Results

- Agglomerative Clustering different methods

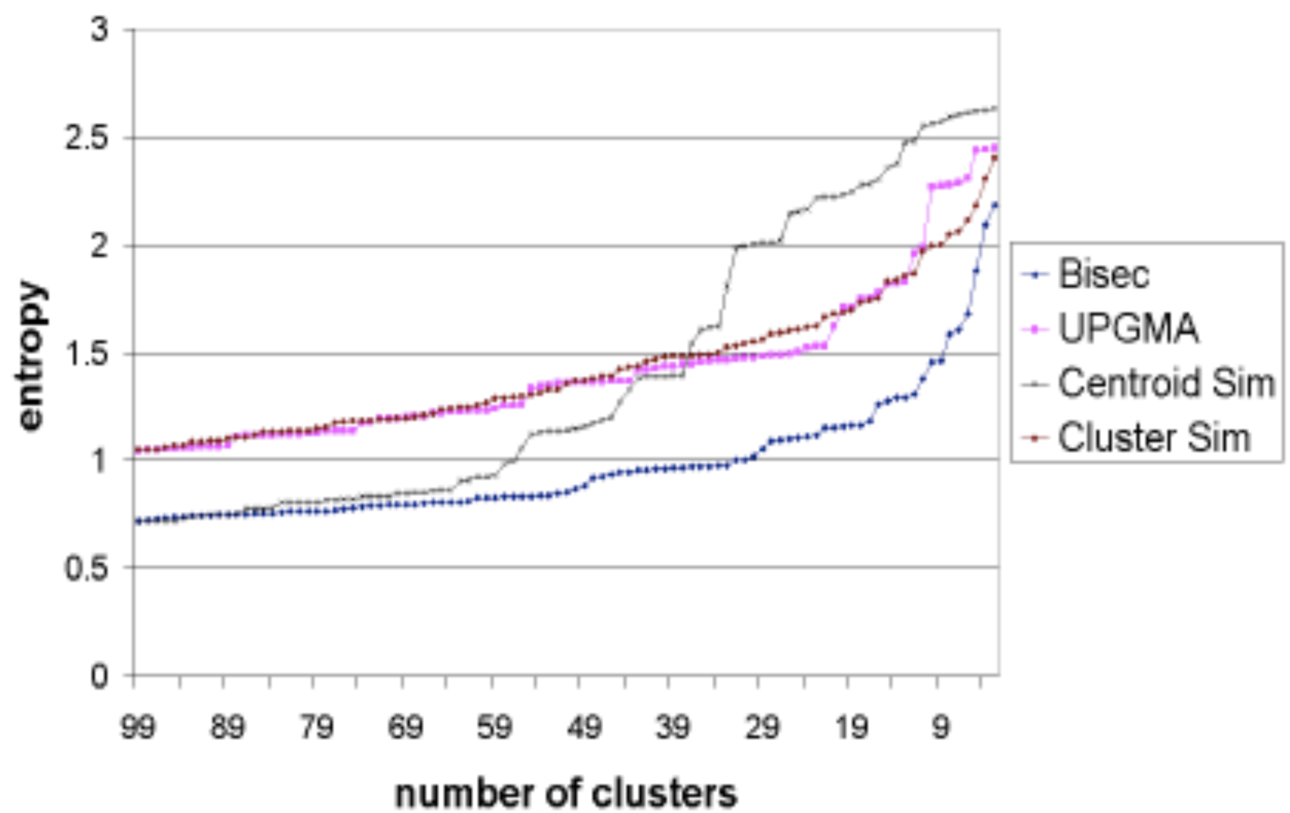


Figure 1: Comparison of entropy for re0 data set.

Steinbach, M., Karypis, G., & Kumar, V. (2000, August). A comparison of document clustering techniques. In KDD workshop on text mining (Vol. 400, No. 1, pp. 525-526).

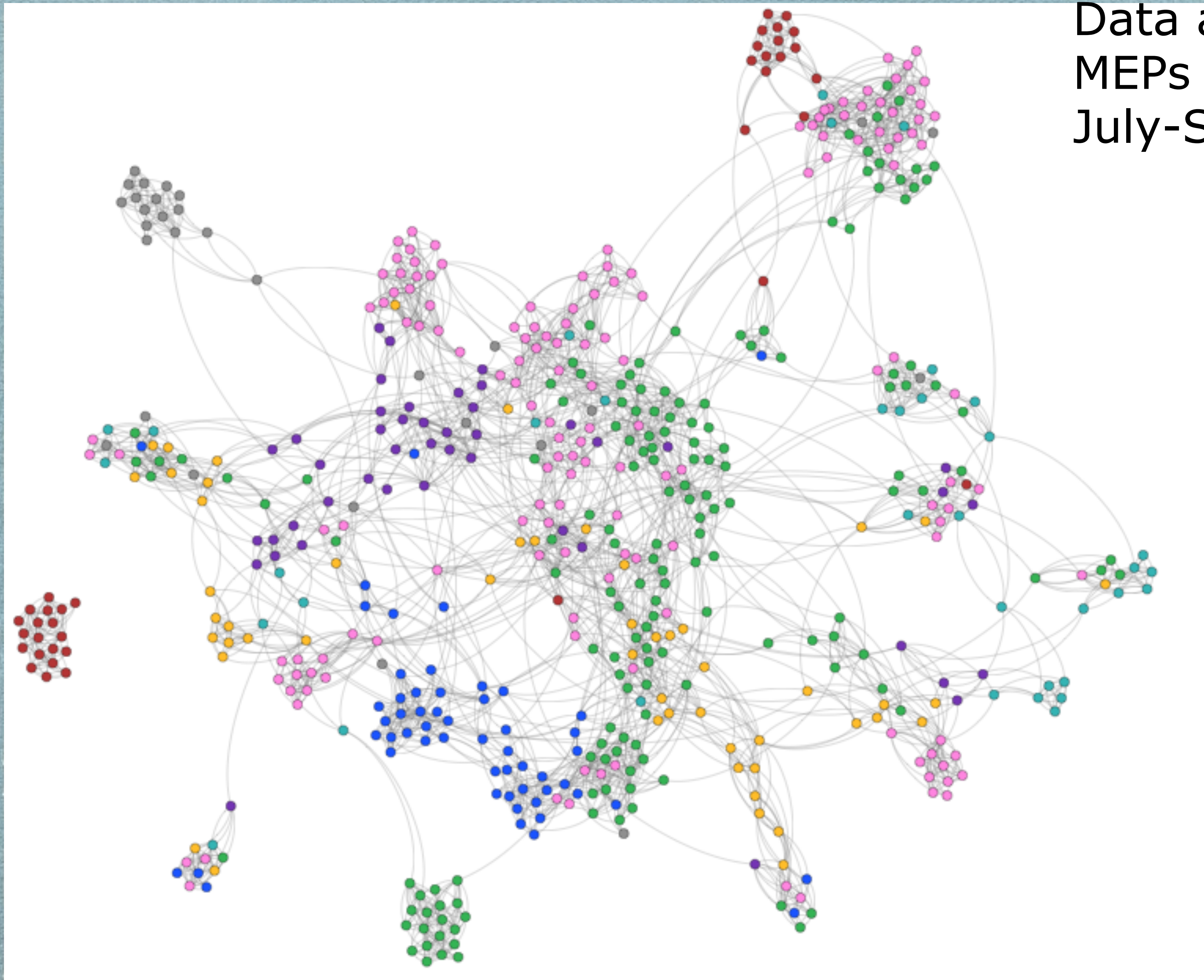
Clustering

Graph-Based Clustering

Graphs: The Idea

- Many problems can be cast as graphs; entities and relations as nodes with links (vertices and edges)
- So, we link up entities on the basis that they are related in some way (mailing, replying, sim docs)
- Then the structure of that graph can tell us about clusters of entities; entities that are "close" or "highly connected" or "related"
- Graph-based clustering uses many ideas from graph theory to define groups

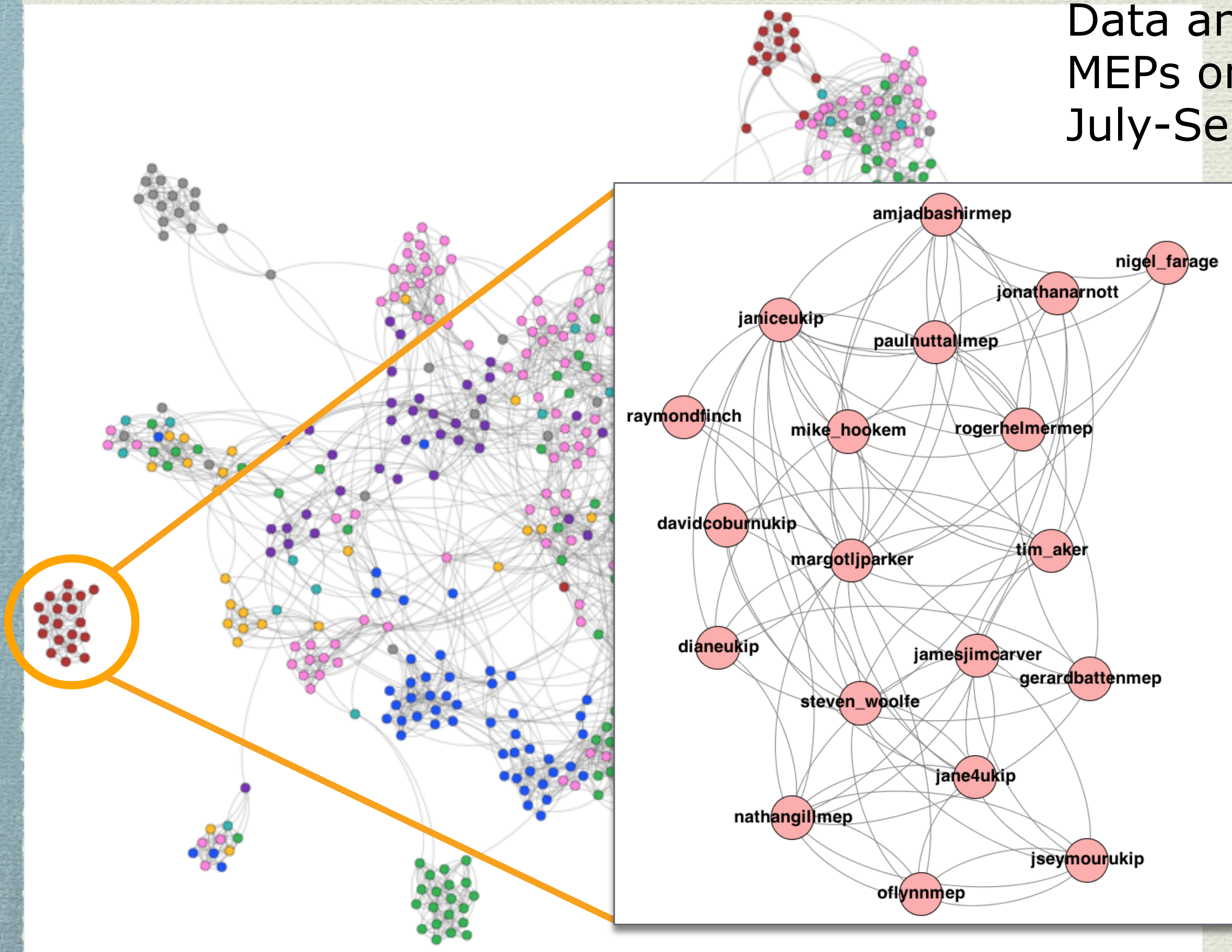
Graphs: The Idea



Data analysed for 584 MEPs on Twitter during July-Sept 2014.

Cross & Greene (2014)

Graphs: The Idea

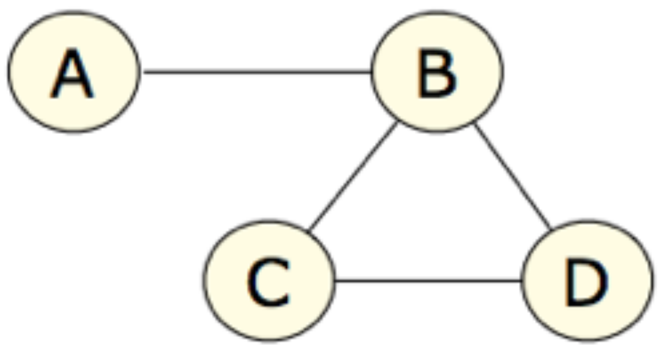


Data analysed for 584 MEPs on Twitter during July-Sept 2014.

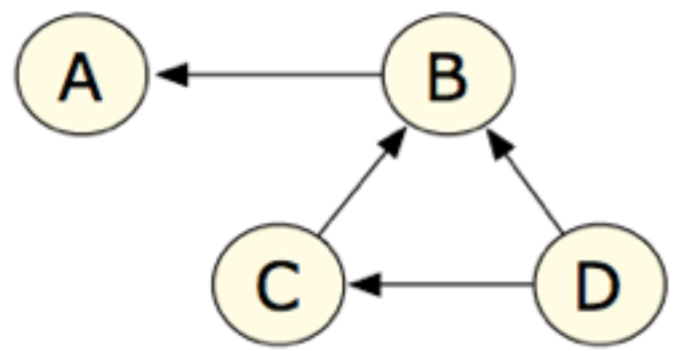
Cross & Greene (2014)

Graphs: Basics I

- Graph: a way of representing the relationships among a collection of objects.
- Consists of a set of objects, called nodes, with certain pairs of these objects connected by links called edges.



Undirected Graph



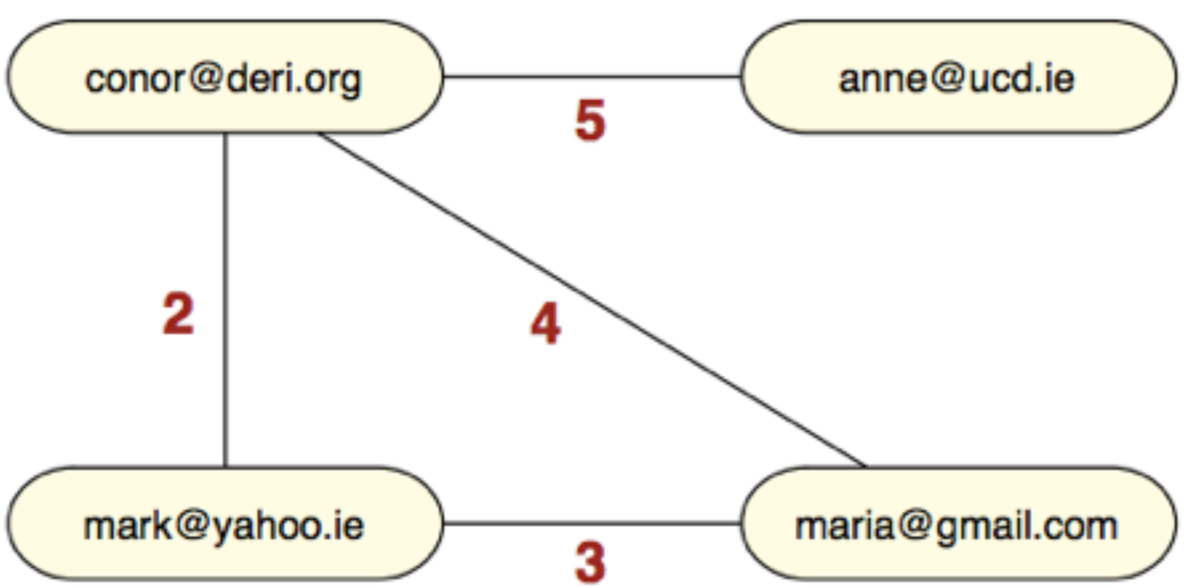
Directed Graph

- Two nodes are neighbours if they are connected by an edge.
- Degree of a node is the number of edges ending at that node.
- For a directed graph, the in-degree and out-degree of a node refer to numbers of edges incoming to or outgoing from the node.

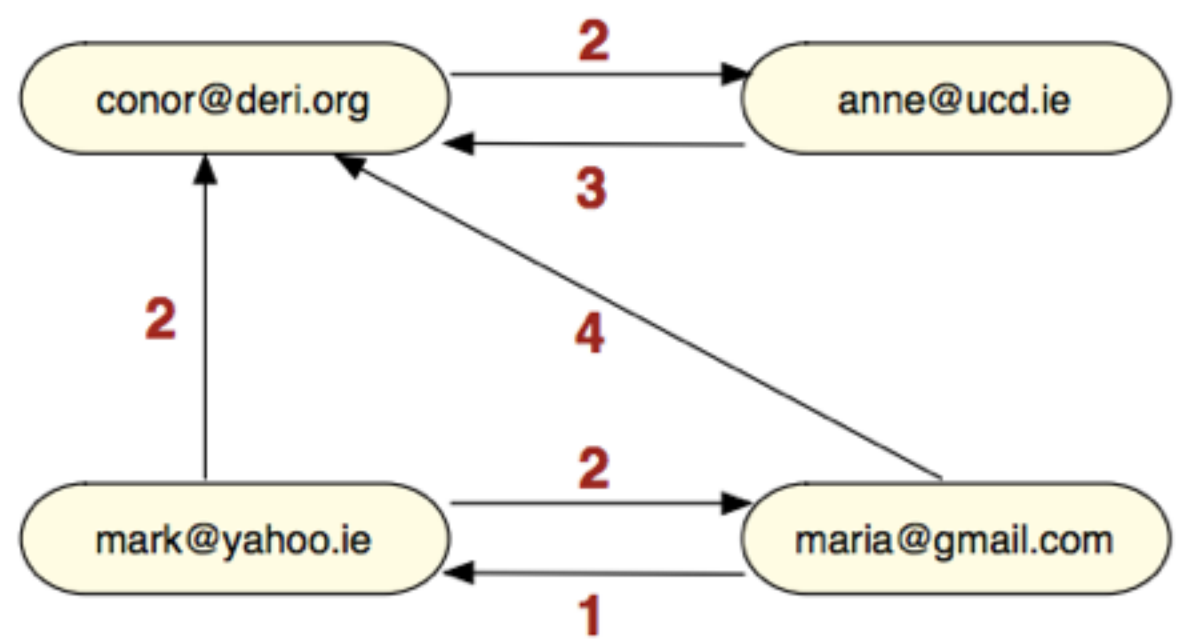
Greene (2011) http://mlg.ucd.ie/files/summer/tutorial.pdf

Graphs: Basics II

- Weighted graph: numeric value is associated with each edge.
- Edge weights may represent a concept such as similarity, distance, or connection cost.



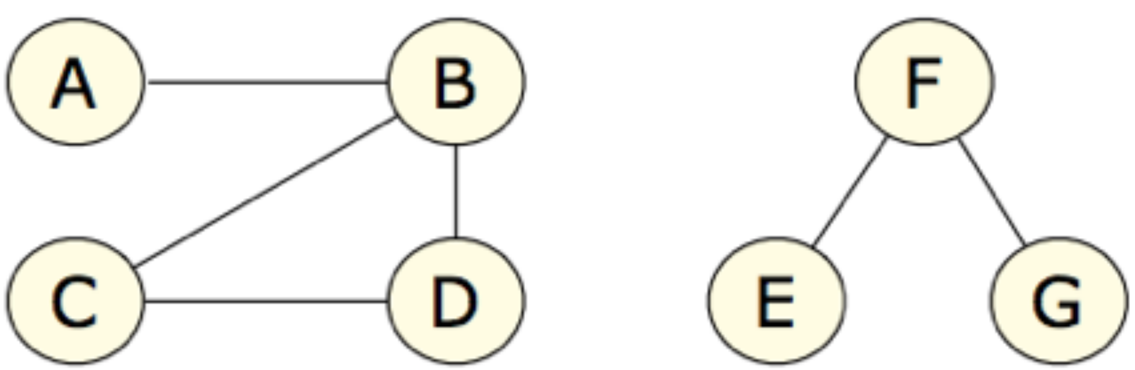
Undirected weighted graph



Directed weighted graph

Graphs: Connectivity & Components

- A graph is connected if there is a path between every pair of nodes in the graph.
- A connected component is a subset of the nodes where:
  1. A path exists between every pair in the subset.
  2. The subset is not part of a larger set with the above property.

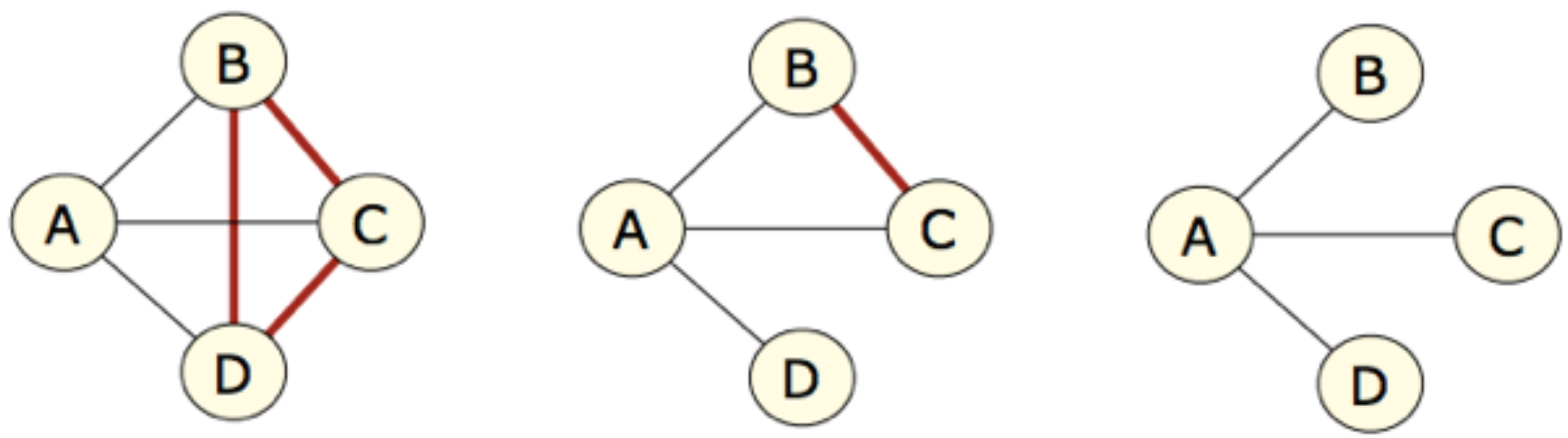


3 connected components

- In many empirical social networks a larger proportion of all nodes will belong to a single giant component.

Graphs: Clustering Coefficient

- The neighbourhood of a node is set of nodes connected to it by an edge, not including itself.
- The clustering coefficient of a node is the fraction of pairs of its neighbours that have edges between one another.

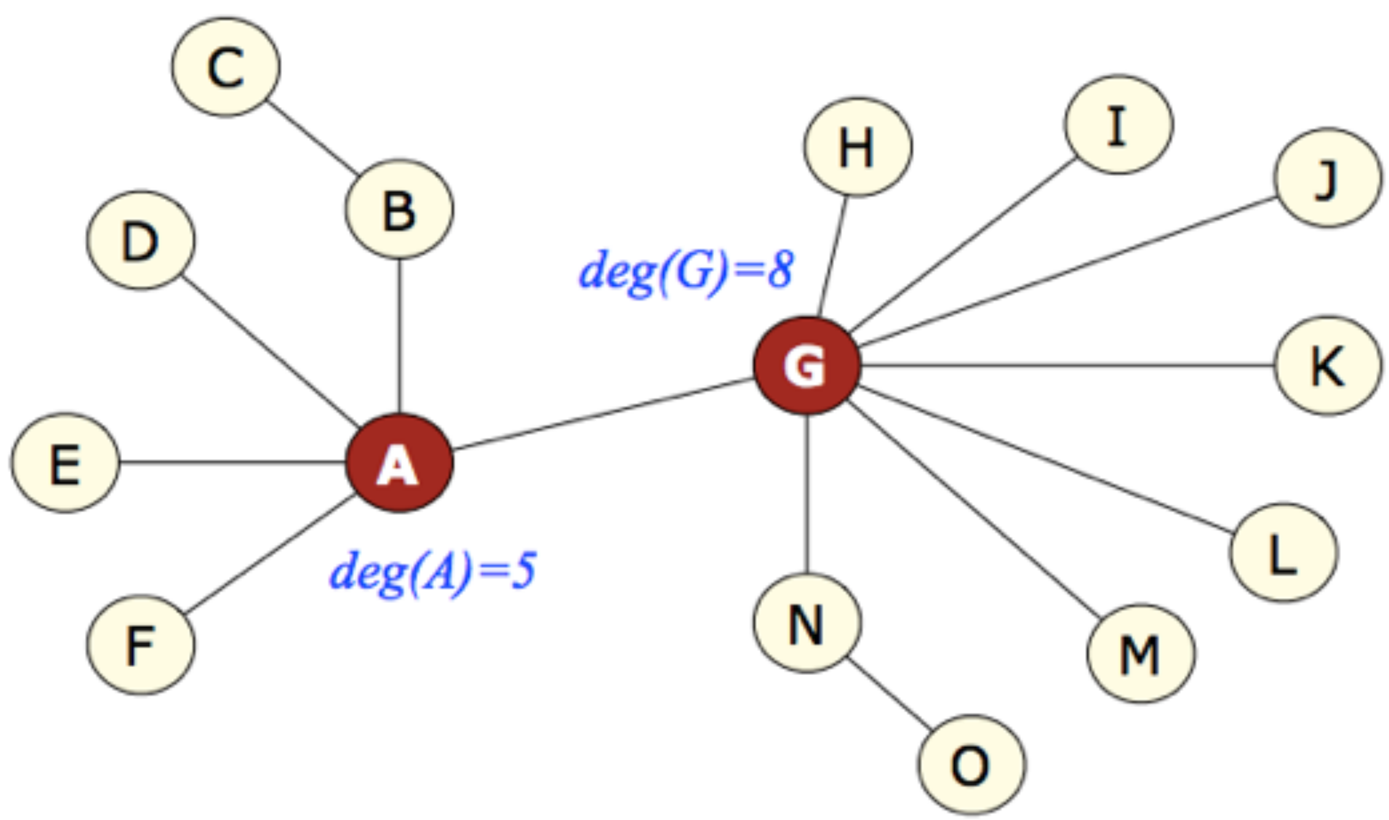


Node A: $CC = \frac{3}{3}$ $CC = \frac{1}{3}$ $CC = \frac{0}{3}$

- Locally indicates how concentrated the neighbourhood of a node is, globally indicates level of clustering in a graph.
- Global score is average over all nodes: $\bar{CC} = \frac{1}{n}\sum_{i=1}^{n} CC(v_i)$

Graphs: Community Detection

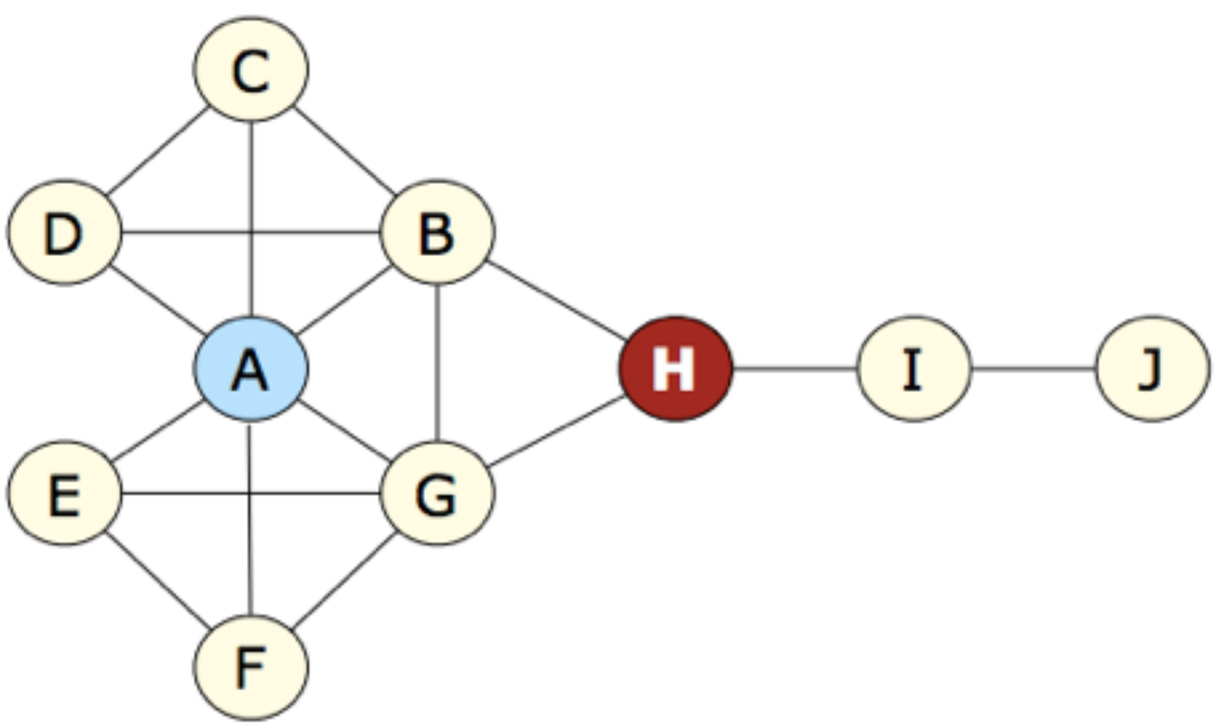
- A variety of different measures exist to measure the importance, popularity, or social capital of a node in a social network.
- Degree centrality focuses on individual nodes - it simply counts the number of edges that a node has.
- Hub nodes with high degree usually play an important role in a network. For directed networks, in-degree is often used as a proxy for popularity.



Graphs: Betweenness Centrality

- A path in a graph is a sequence of edges joining one node to another. The path length is the number of edges.
- Often want to find the shortest path between two nodes.
- A graph's diameter is the longest shortest path over all pairs of nodes.

- Nodes that occur on many shortest paths between other nodes in the graph have a high betweenness centrality score.



Node "A" has high degree centrality than "B", as "B" has few direct connections.

Node "H" has higher betweenness centrality, as "H" plays a broker role in the network

Graphs: Centrality

- The eigenvector centrality of a node proportional to the sum of the centrality scores of its neighbours.

➡ A node is important if it connected to other important nodes.

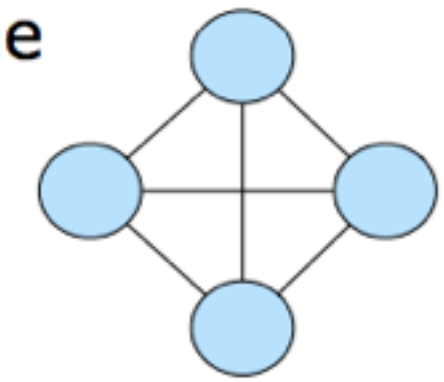
➡ A node with a small number of influential contacts may outrank one with a larger number of mediocre contacts.

- Computation:

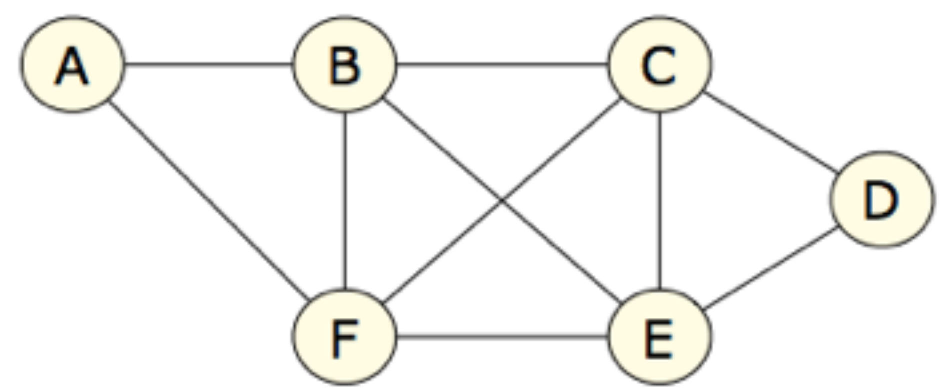
1. Calculate the eigendecomposition of the pairwise adjacency matrix of the graph.
2. Select the eigenvector associated with largest eigenvalue.
3. Element *i* in the eigenvector gives the centrality of the *i*-th node.

Graphs: Cliques

- A clique is a social grouping where everyone knows everyone else (i.e. there is an edge between each pair of nodes).
- A maximal clique is a clique that is not a subset of any other clique in the graph.
- A clique with size greater than or equal to that of every other clique in the graph is called a maximum clique.

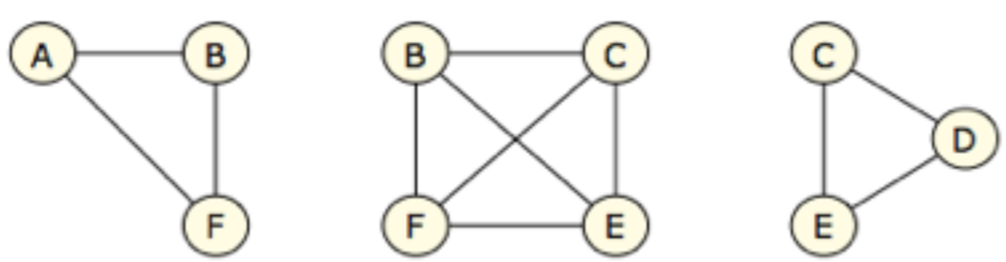


Find all maximal cliques in the specified graph:




```
>>> cl = list( networkx.find_cliques(g) )
```

```
>>> print cl
[['a', 'b', 'f'], ['c', 'e', 'b', 'f'], ['c', 'e', 'd']]
```



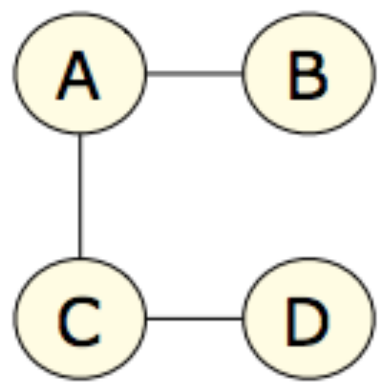
@ Greene (2011)

Graphs: Using Them…

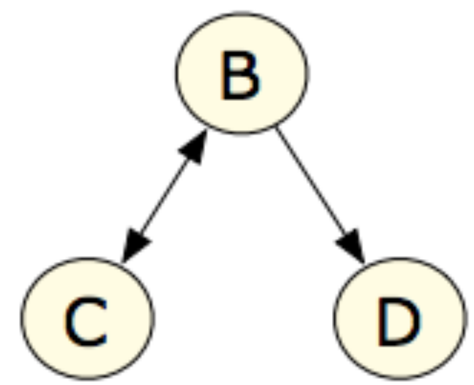
- So, these different properties of the graph are used to find clusters based on how :
  - the graph is structured
  - a node is connected to other nodes
  - one group of nodes differs from another
- Properties like (i) clique numbers, (ii) clique structure, (ii) centrality, (iii) cluster coefficients, (iv) in-/out-degrees, (v) weights/distance

A Clique Means…

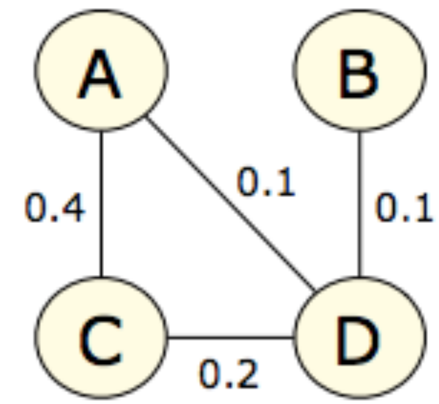
Scientific Research Network



Co-authorship Graph

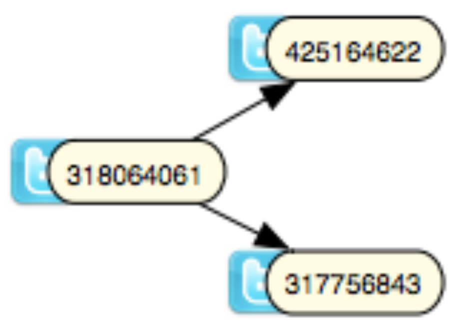


Citation Graph

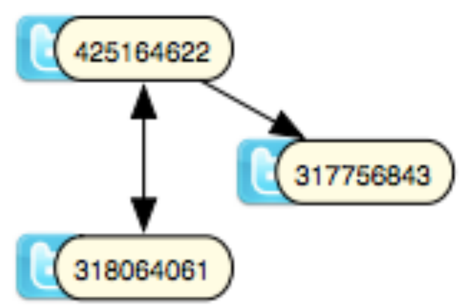


Content Similarity

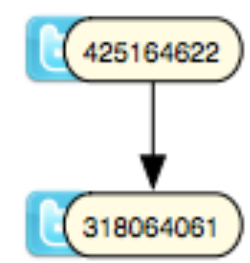
Microblogging Network



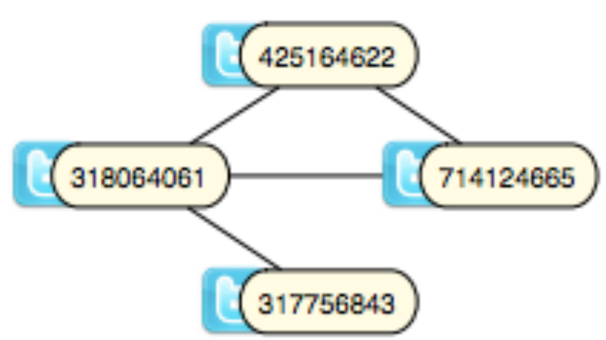
Follower Graph



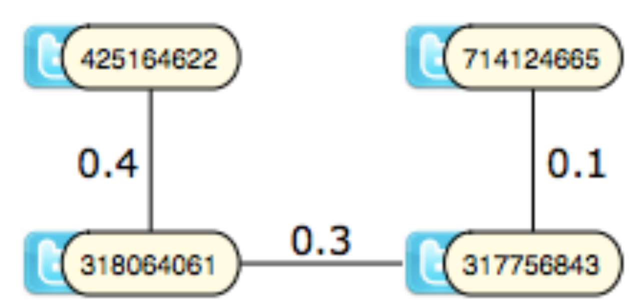
Reply-To Graph



Mention Graph



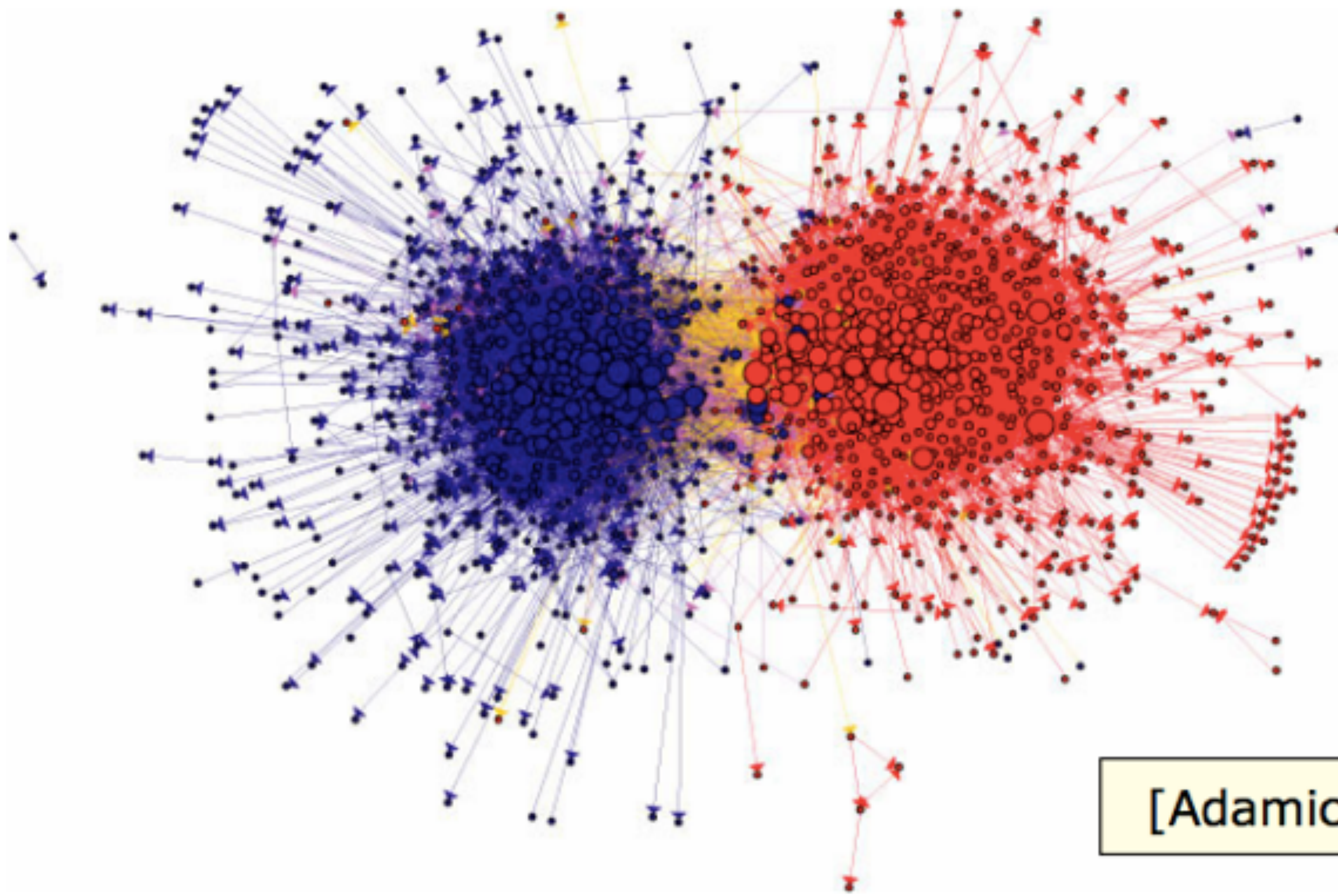
Co-Listed Graph



Content Similarity

Community Detection

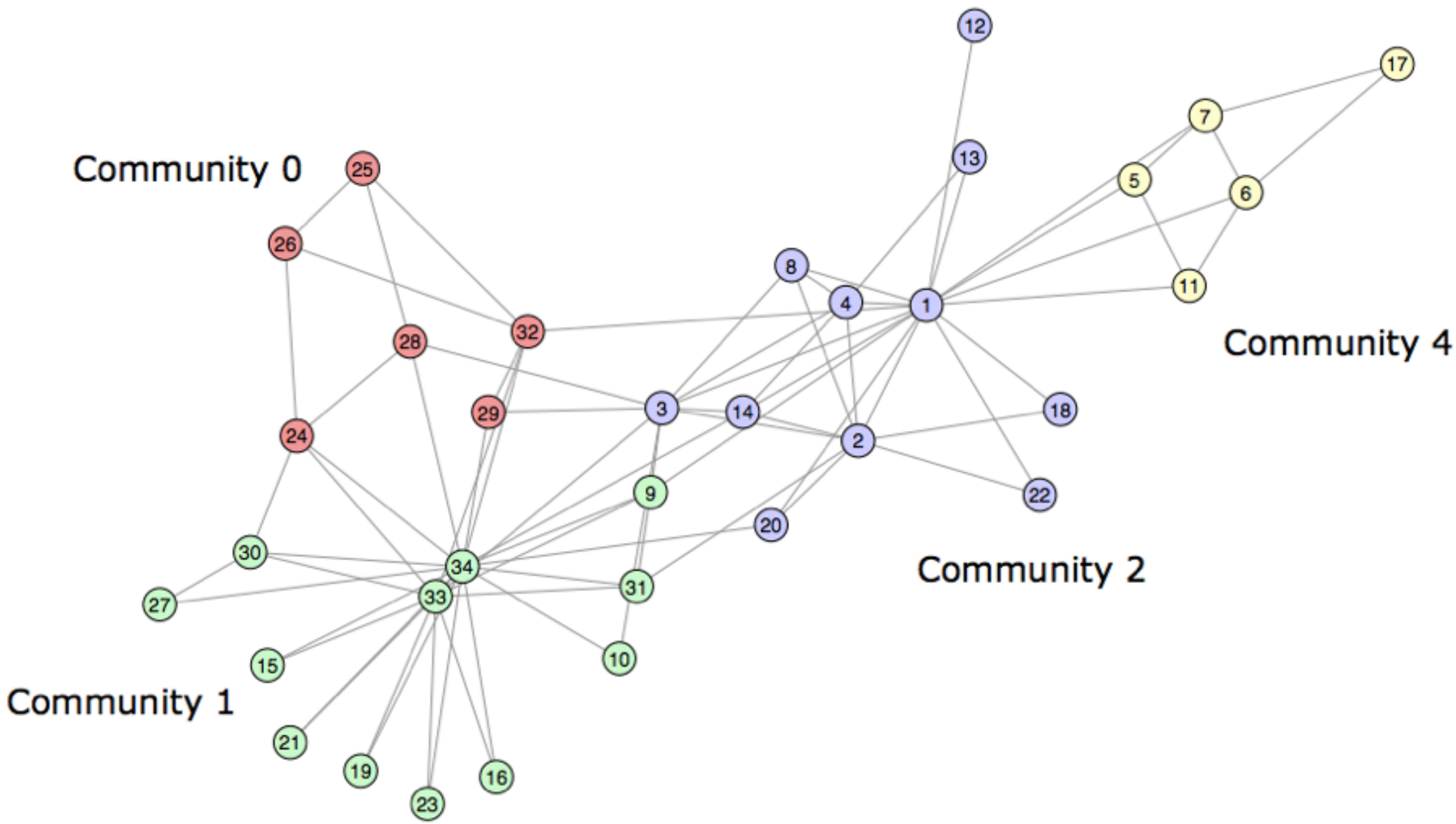
- We will often be interested in identifying communities of nodes in a network...



[Adamic & Glance,2005]

- Example: Two distinct communities of bloggers discussing 2004 US Presidential election.

Social Media Community is…

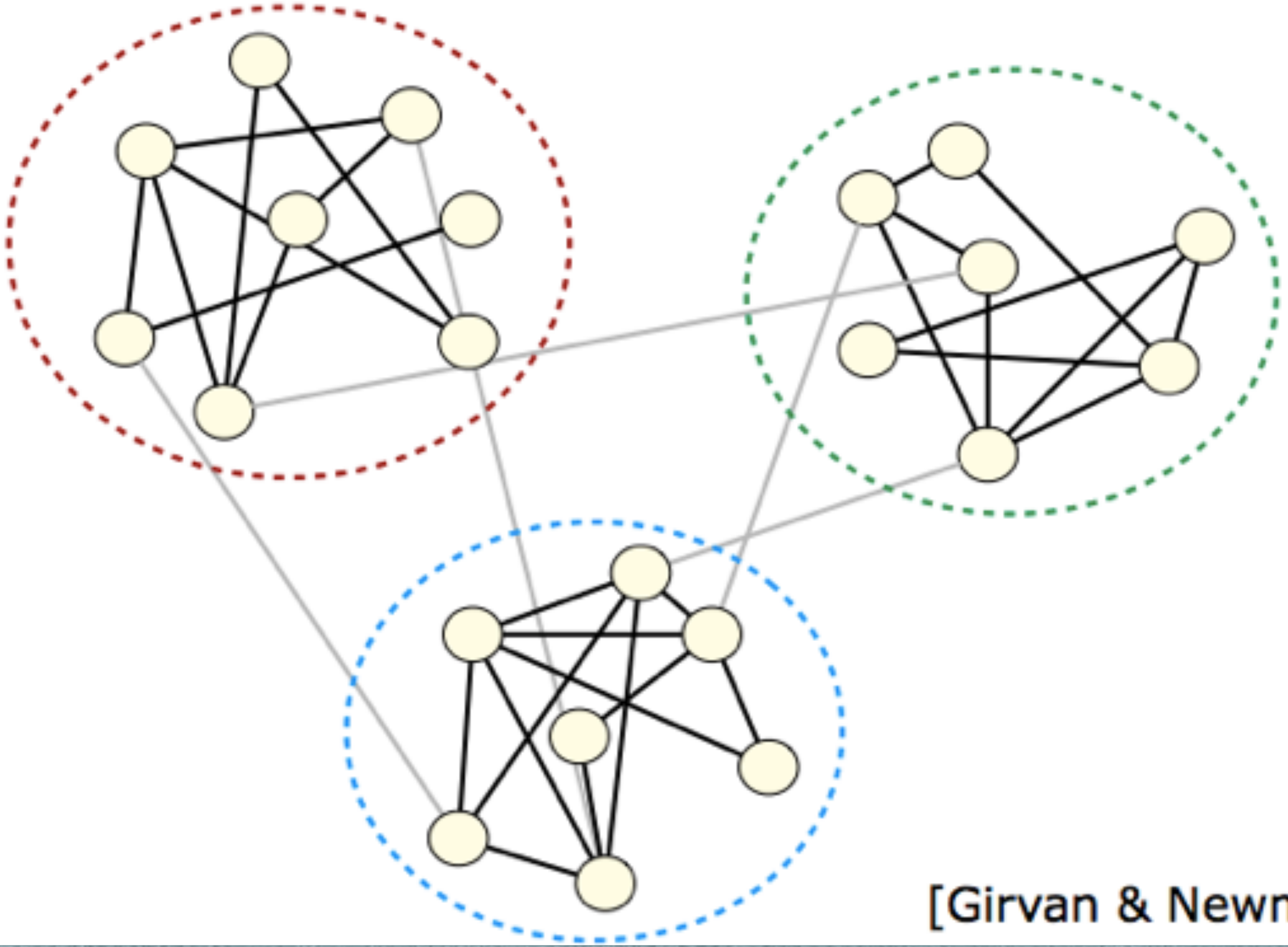


Graphs: Clustering Methods

- So, there are many different methods that can be used to find interesting clusters in graphs:
  - computing centrality (hubs and authorities)
  - finding connected components (connected farms)
  - partition the graph according to some criteria
  - one group of nodes differs from another
- Properties like (i) clique numbers, (ii) clique structure, (ii) centrality, (iii) cluster coefficients, (iv) in-/out-degrees, (v) weights/distance

Graphs: Community Detection

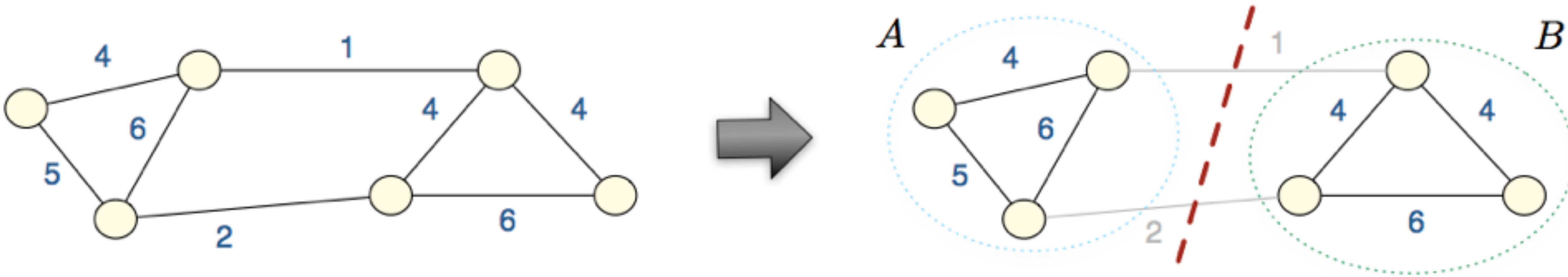
- A variety of definitions of community/cluster/module exist:
  - A group of nodes which share common properties and/or play a similar role within the graph [Fortunato, 2010].
  - A subset of nodes within which the node-node connections are dense, and the edges to nodes in other communities are less dense [Girvan & Newman, 2002].



[Girvan & Newman, 2002]

ClusterTech: Partitioning

- **Goal:** Divide the nodes in a graph into a user-specified number of <u>disjoint</u> groups to optimise a criterion related to number of edges cut.



- Min-cut simply involves minimising number (or weight) of edges cut by the partition.

$$cut(A, B) = 3$$

- Recent approaches use more sophisticated criteria (e.g. normalised cuts) and apply multi-level strategies to scale to large graphs.

Graclus [Dhillon et al, 2007] http://www.cs.utexas.edu/users/dml/Software/graclus.html

<u>Issues:</u> Requirement to pre-specify number of partitions, cut criteria often make strong assumptions about cluster structur

CTech: Modularity Optimisation

- Newman & Girvan [2004] proposed measure of partition quality....
  - ➡ Random graph shouldn't have community structure.
  - ➡ Validate existence of communities by comparing actual edge density with expected edge density in random graph.

$$Q = (\text{number of edges within communities}) - (\text{expected number within communities})$$

- Apply agglomerative technique to iteratively merge groups of nodes to form larger communities such that modularity increases after merging.
- Recently efficient greedy approaches to modularity maximisation have been developed that scale to graphs with up to 10^9 edges.

Louvain Method [Blondel et al, 2008] http://findcommunities.googlepages.com

<u>Issues for Community Detection:</u>

- Total number of edges in graph controls the resolution at which communities are identified [Fortunato, 2010].
- Is it realistic/useful to assign nodes to only a single community?

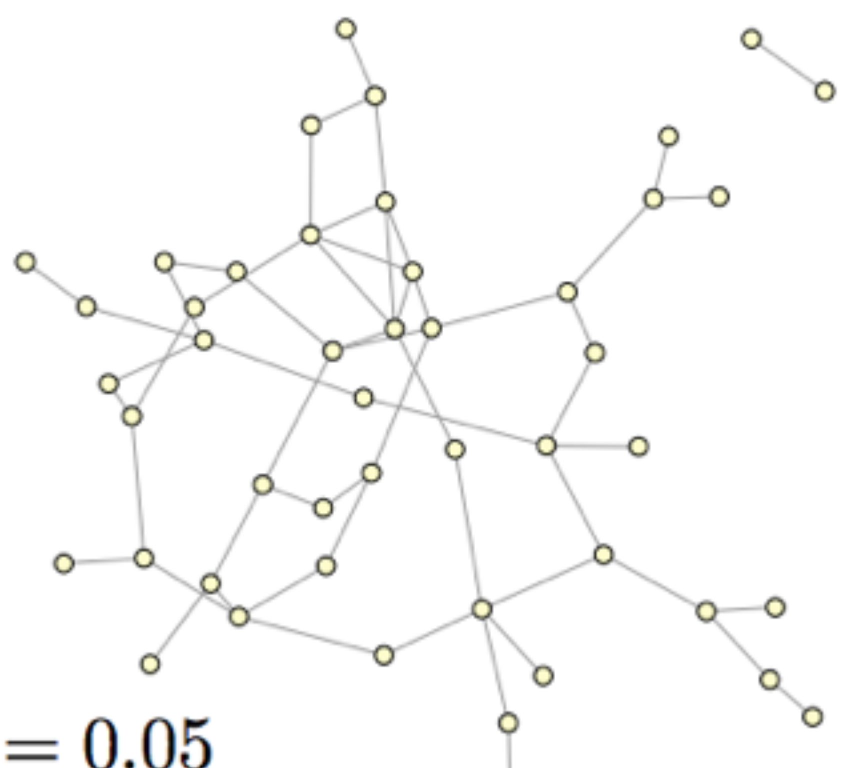
Graphs: Random Networks

- Erdős–Rényi random graph model:
  - Start with a collection of *n* disconnected nodes.
  - Create an edge between each pair of nodes with a probability *p*, independently of every other edge.

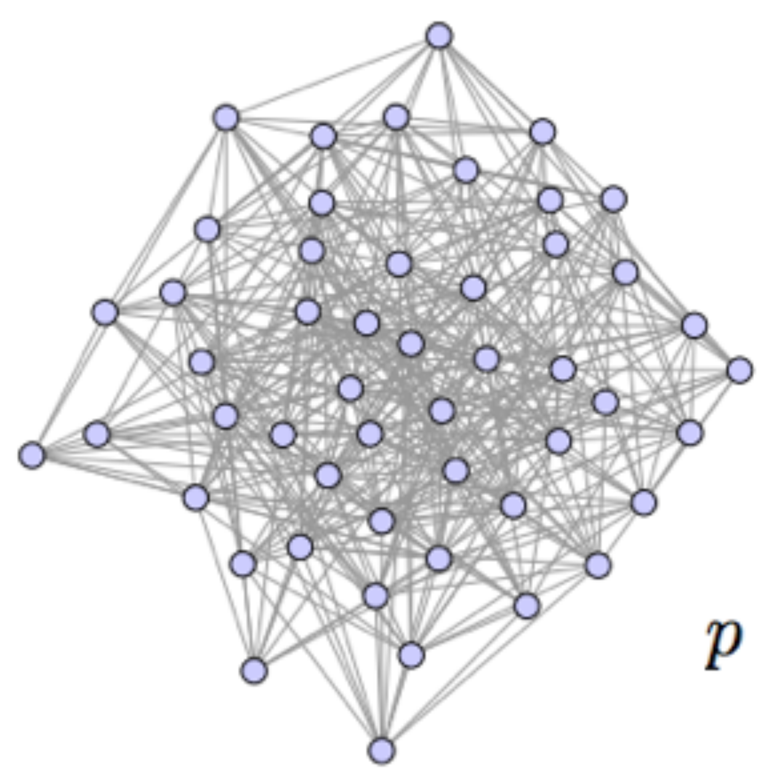
```
g1 = networkx.erdos_renyi_graph(50, 0.05)
```

```
g2 = networkx.erdos_renyi_graph(50, 0.3)
```

Specify number of nodes to create, and connection probability *p*.



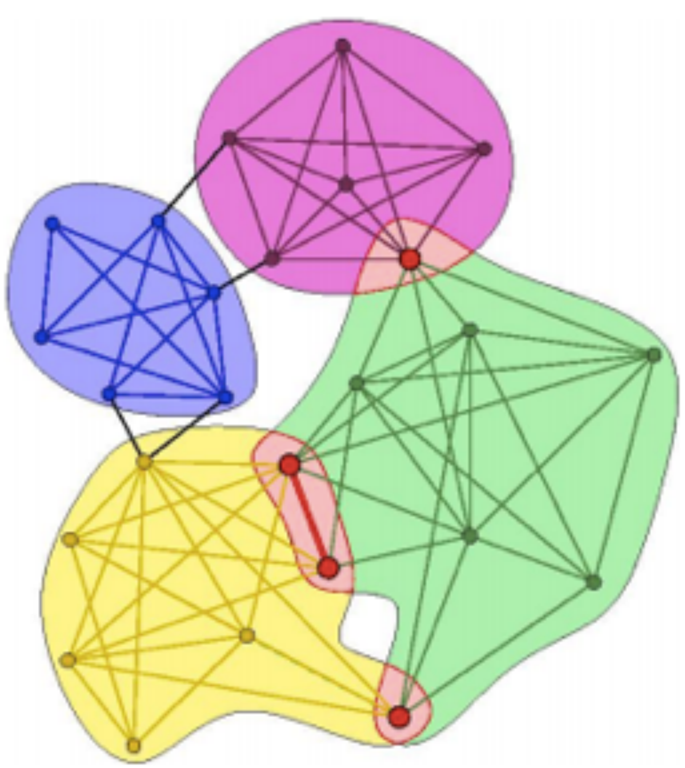
$p = 0.05$



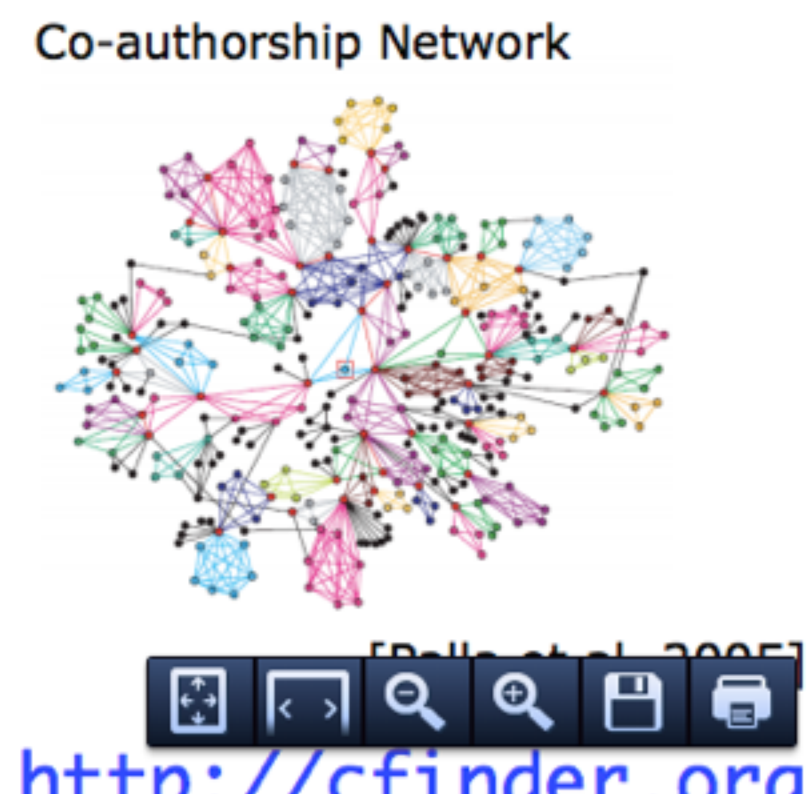
$p = 0.3$

k-cliques & overlapping communities

- **CFinder**: algorithm based on the clique percolation method [Palla et al, 2005].
- Identify *k*-cliques: a fully connected subgraph *k* nodes.
- Pair of *k*-cliques are "adjacent" if they share *k*−1 nodes.
- Form overlapping communities from maximal union of *k*-cliques that can be reached from each other through adjacent *k*-cliques.



Set of overlapping communities built from 4-cliques.



Graphs: Caution

- Note that a lot of graph-research is not specifically or uniquely about text analytics
- More about social relations; but there are cases where the graph relations are based on analysing the text of nodes (eg. MEP)
- We consider egs that are mainly textual

EG1: Kleinberg's Memes

- Looks at how memes (quotes) rise and fall with news attention…"lipstick on a pig"…analyses 90M item corpus of quotes in news articles & blogs
- Built graphs of the links between quote-bits (with frequencies) and partitioned them into components by deleting links in the graph
- Then pictured changing occurrences over time

Leskovec, J., Backstrom, L., & Kleinberg, J. (2009, June). Meme-tracking and the dynamics of the news cycle. In 15th ACM SIGKDD (pp. 497-506). ACM.

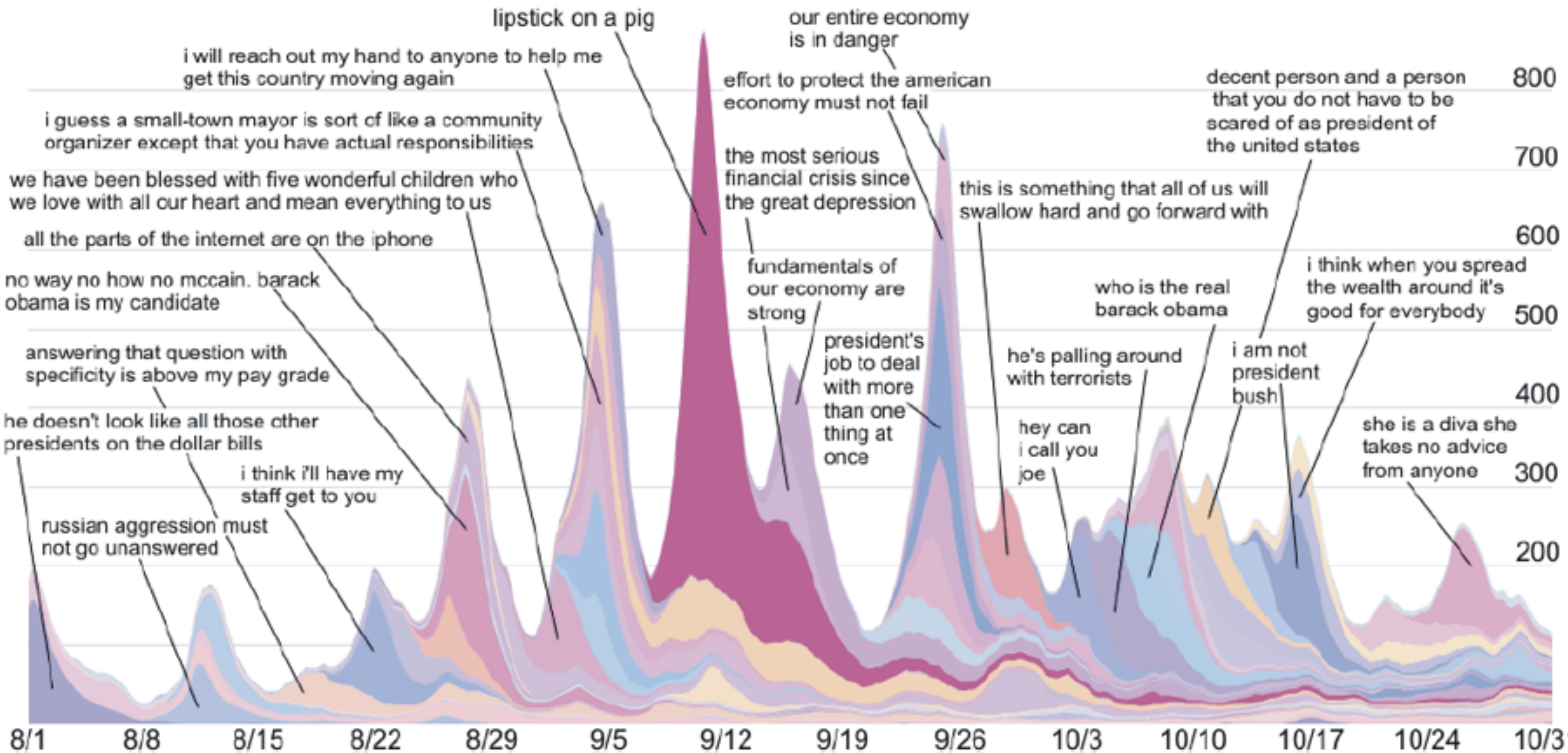


Figure 4: Top 50 threads in the news cycle with highest volume for the period Aug. 1 – Oct. 31, 2008. Each thread consists of all news articles and blog posts containing a textual variant of a particular quoted phrases. (Phrase variants for the two largest threads in each week are shown as labels pointing to the corresponding thread.) The data is drawn as a stacked plot in which the thickness of the strand corresponding to each thread indicates its volume over time. Interactive visualization is available at http://memetracker.org.

Leskovec, J., Backstrom, L., & Kleinberg, J. (2009, June). Meme-tracking and the dynamics of the news cycle. In 15th ACM SIGKDD (pp. 497-506). ACM.

EG1: Kleinberg's Memes

- To find clusters of phrases from news, that should be quotes ("genetic signatures"); but note how they are not always identical !
- Corpus 90M news articles and blogs from Q4 2008; 390GB; 122M quotes, cut down to 47M common ones
- Build a phrase graph, where each node is a phrase and directed edges connect "related" phrases

Leskovec, J., Backstrom, L., & Kleinberg, J. (2009, June). Meme-tracking and the dynamics of the news cycle. In 15th ACM SIGKDD (pp. 497-506). ACM.

want all phrases “belonging”to a long quote

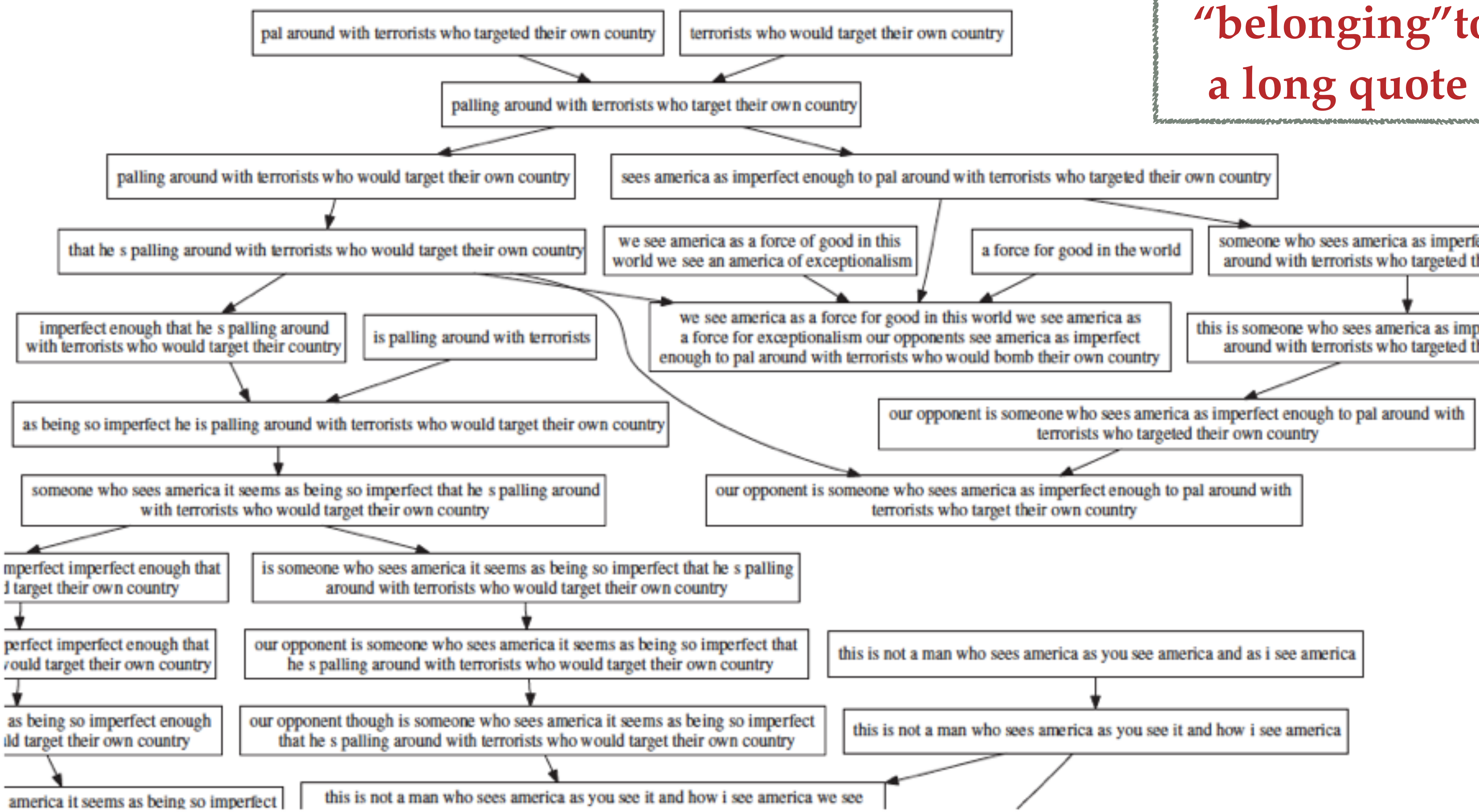


Figure 1: A small portion of the full set of variants of Sarah Palin’s quote, “Our opponent is someone who sees America, it seems, as being so imperfect, imperfect enough that he’s palling around with terrorists who would target their own country.” The arrows indicate the (approximate) inclusion of one variant in another, as part of the methodology developed in Section 2.

Leskovec, J., Backstrom, L., & Kleinberg, J. (2009, June). Meme-tracking and the dynamics of the news cycle. In 15th ACM SIGKDD (pp. 497-506). ACM.

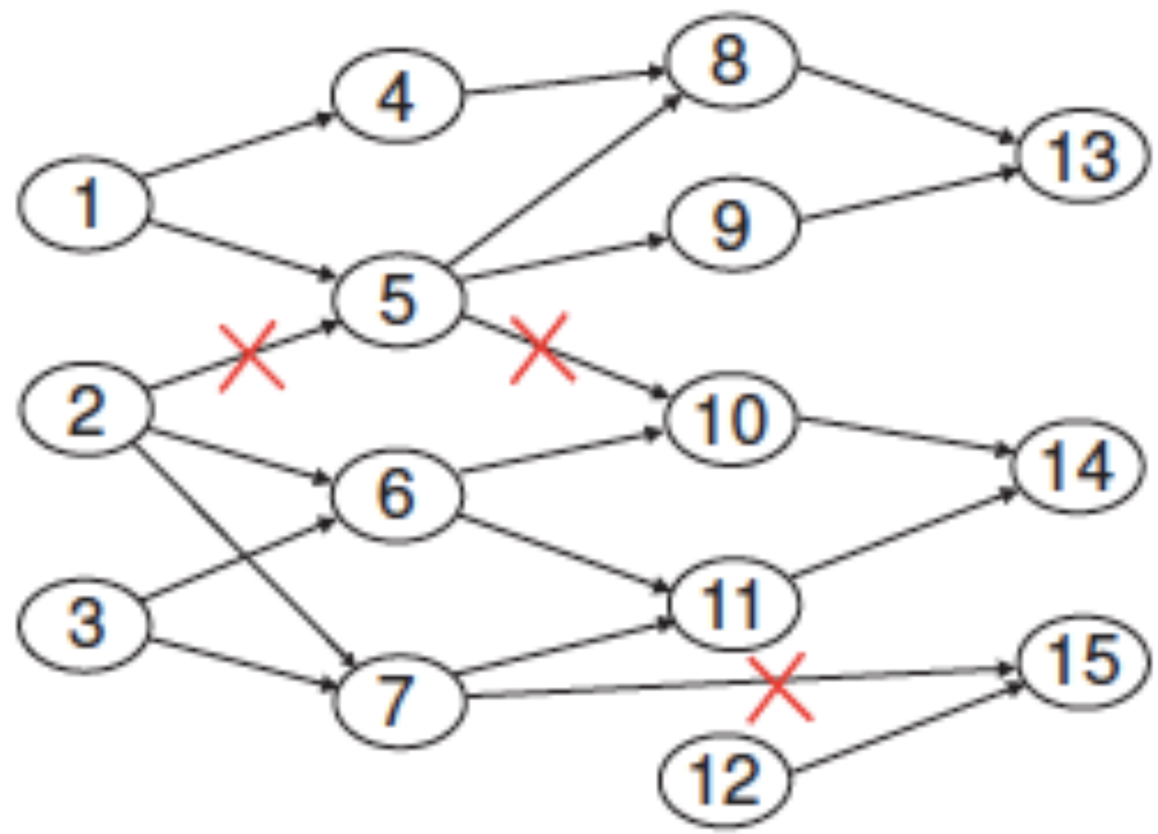


Figure 2: Phrase graph. Each phrase is a node and we want to delete the least edges so that each resulting connected component has a single root node/phase, a node with zero out-edges. By deleting the indicated edges we obtain the optimal solution.

Leskovec, J., Backstrom, L., & Kleinberg, J. (2009, June). Meme-tracking and the dynamics of the news cycle. In 15th ACM SIGKDD (pp. 497-506). ACM.

EG1: Phase Selection

- Phrase is quoted string in a given article
- All quotes: lower-bound, *L*, on phrase word-length
- Lower-bound, *M*, on frequency of phrase in corpus
- Eliminate phrases with *e* fraction of item occurrences from the same domain (aka Spammers)
- Used e=.25, M = 10, L = 4

Leskovec, J., Backstrom, L., & Kleinberg, J. (2009, June). Meme-tracking and the dynamics of the news cycle. In 15th ACM SIGKDD (pp. 497-506). ACM.

EG1: Graph Building & Cuts

- Build graph: *p* and *q* are related when *p* is contiguous subsequence of the words in phrase *q*, with tolerance for mismatching (edit distance = 1; 10 in-a-row)
- All edges in this DAG, point from shorter *p*-phrases to longer *q*-phrases, with weights on edges that reflect edit-distance between *p* and *q* plus frequency of *q*
- Partition graph by cutting edges; phrase cluster should be a subgraph for which all paths (*p*s and minor *q*s) terminate in a single root node (*q*), delete edges of small total weight from phrase graph so it falls apart into disjoint parts
- Clustering becomes a <u>*DAG partitioning problem*</u> turns out to be NP-hard for optimal solutions but they do a heuristic version that works well

> DAG Partitioning: Given a directed acyclic graph with edge weights, delete a set of edges of minimum total weight so that each of the resulting components is single-rooted.

Leskovec, J., Backstrom, L., & Kleinberg, J. (2009, June). Meme-tracking and the dynamics of the news cycle. In 15th ACM SIGKDD (pp. 497-506). ACM.

want all phrases "belonging"to a long quote

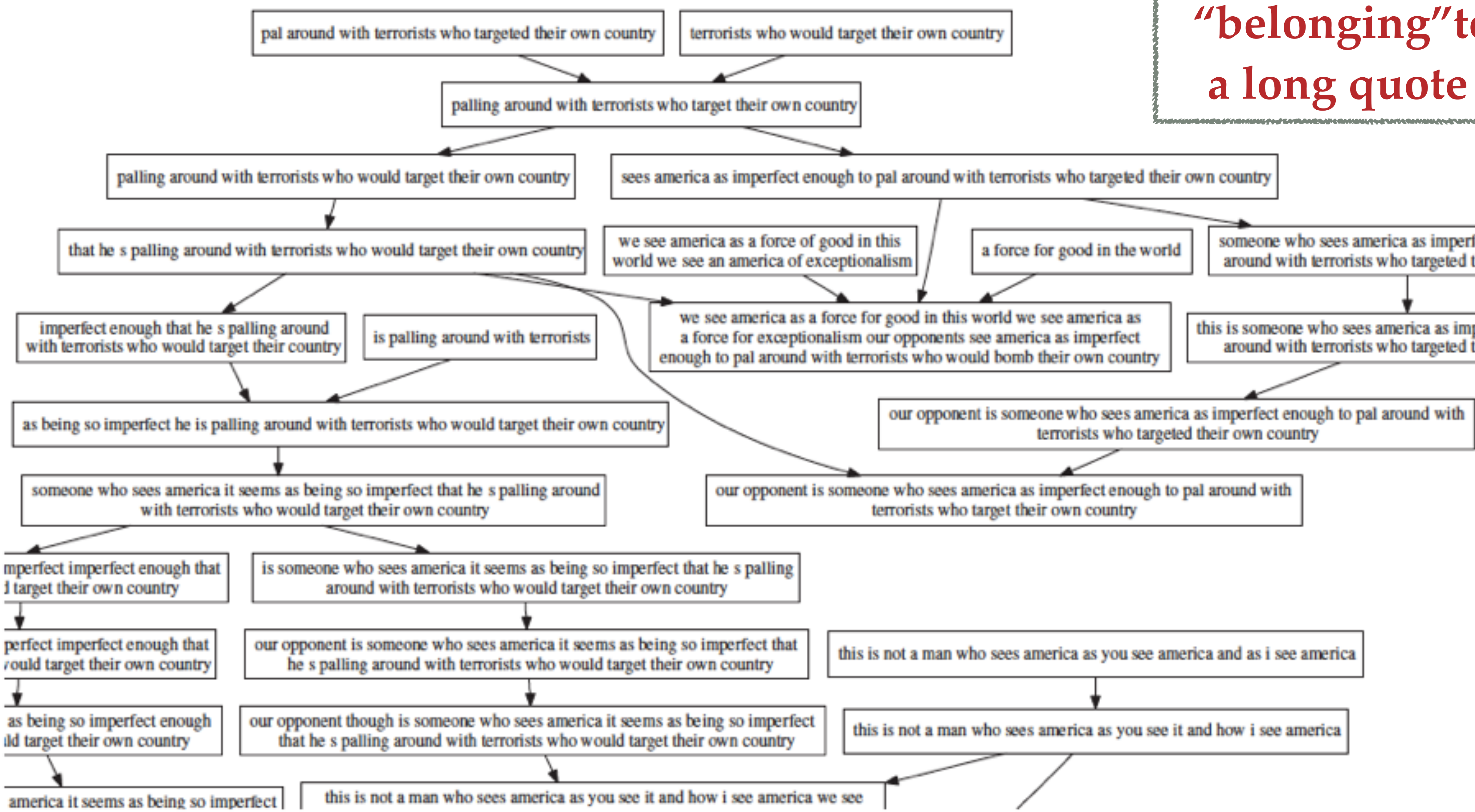


Figure 1: A small portion of the full set of variants of Sarah Palin's quote, "Our opponent is someone who sees America, it seems, as being so imperfect, imperfect enough that he's palling around with terrorists who would target their own country." The arrows indicate the (approximate) inclusion of one variant in another, as part of the methodology developed in Section 2.

Leskovec, J., Backstrom, L., & Kleinberg, J. (2009, June). Meme-tracking and the dynamics of the news cycle. In 15th ACM SIGKDD (pp. 497-506). ACM.

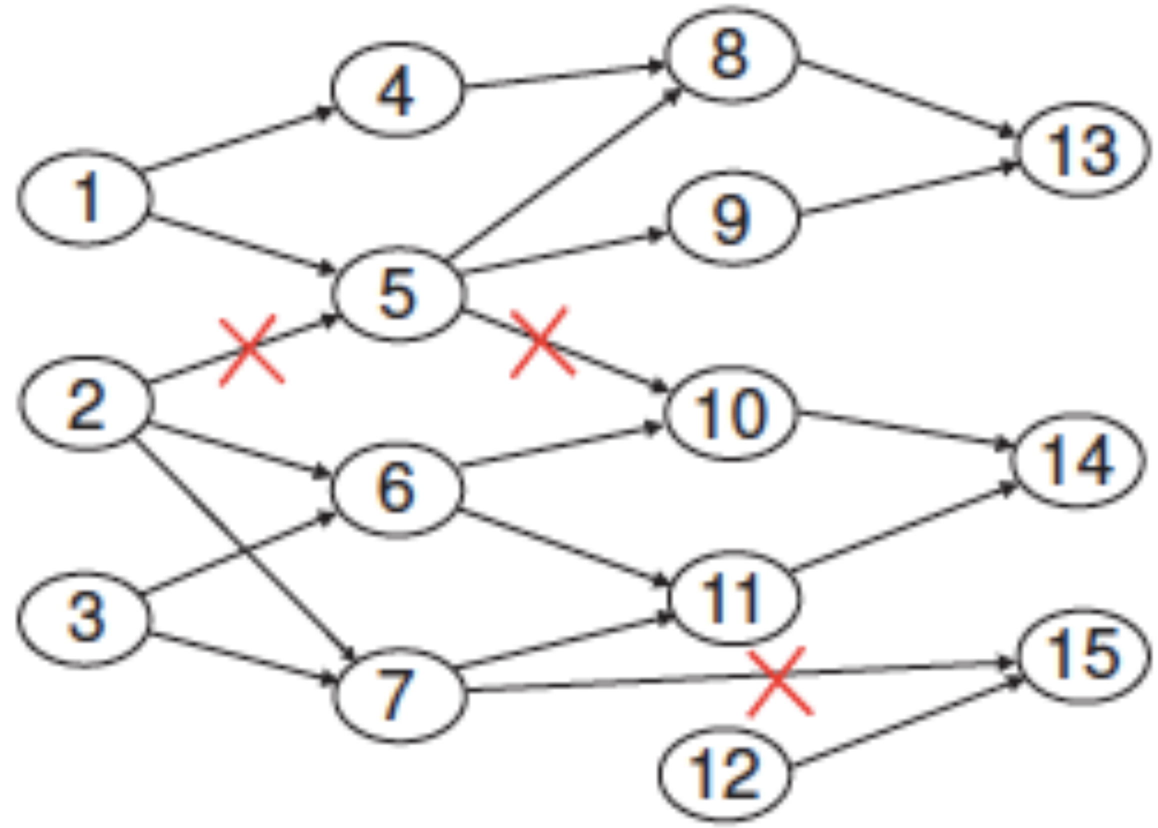


Figure 2: Phrase graph. Each phrase is a node and we want to delete the least edges so that each resulting connected component has a single root node/phase, a node with zero out-edges. By deleting the indicated edges we obtain the optimal solution.

Leskovec, J., Backstrom, L., & Kleinberg, J. (2009, June). Meme-tracking and the dynamics of the news cycle. In 15th ACM SIGKDD (pp. 497-506). ACM.

122M quotes reduces to 47M phrases (22M distinct)

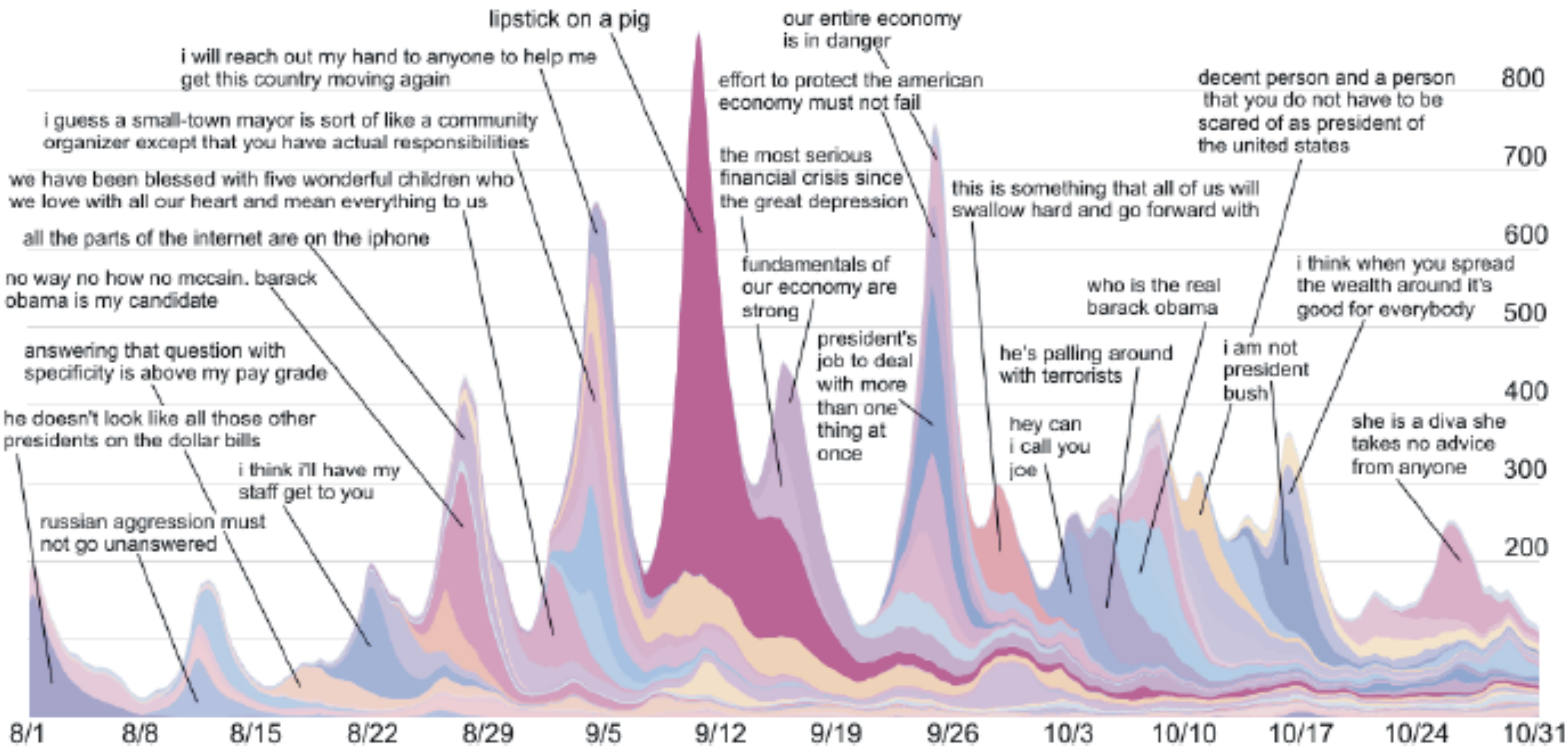


Figure 4: Top 50 threads in the news cycle with highest volume for the period Aug. 1 – Oct. 31, 2008. Each thread consists of all news articles and blog posts containing a textual variant of a particular quoted phrases. (Phrase variants for the two largest threads in each week are shown as labels pointing to the corresponding thread.) The data is drawn as a stacked plot in which the thickness of the strand corresponding to each thread indicates its volume over time. Interactive visualization is available at http://memetracker.org.

Leskovec, J., Backstrom, L., & Kleinberg, J. (2009, June). Meme-tracking and the dynamics of the news cycle. In 15th ACM SIGKDD (pp. 497-506). ACM.

Shows the “dynamic of the news cycle”: the shape of the increase in attention on a story, and it drop off; difference ‘tween news and blogs; that news pre-dates blogs

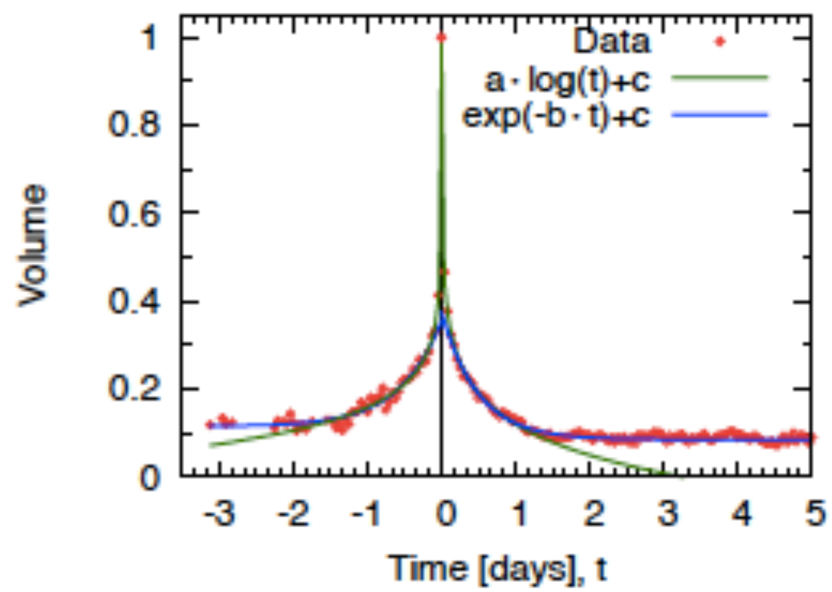


Figure 7: Thread volume increase and decay over time. Notice the symmetry, quicker decay than buildup, and lower baseline popularity after the peak.

Rank	Lag [h]	Reported	Site
1	-26.5	42	hotair.com
2	-23	33	talkingpointsmemo.com
4	-19.5	56	politicalticker.blogs.cnn.com
5	-18	73	huffingtonpost.com
6	-17	49	digg.com
7	-16	89	breitbart.com
8	-15	31	thepoliticalcarnival.blogspot.com
9	-15	32	talkleft.com
10	-14.5	34	dailykos.com
16	-14	54	blogs.abcnews.com
30	-11	32	uk.reuters.com
34	-11	72	cnn.com
40	-10.5	78	washingtonpost.com
48	-10	53	online.wsj.com
49	-10	54	ap.org

Table 1: How quickly different media sites report a phrase. Lag: median time between the first mention of a phrase on a site and the time when its mentions peaked. Reported: percentage of top 100 phrases that the site mentioned.

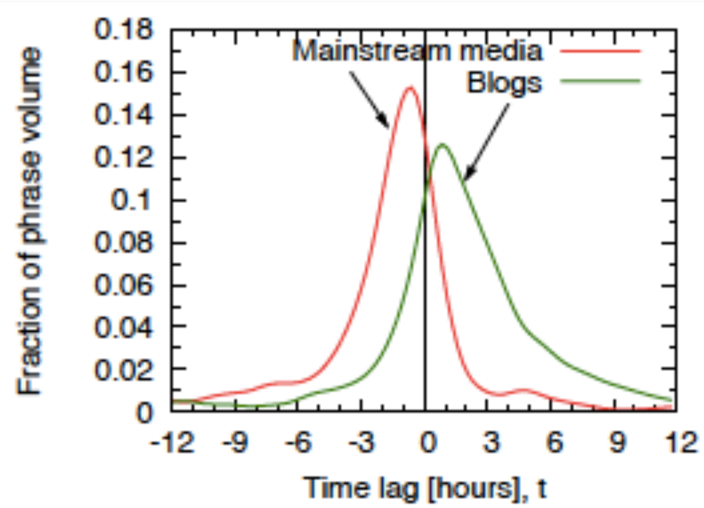
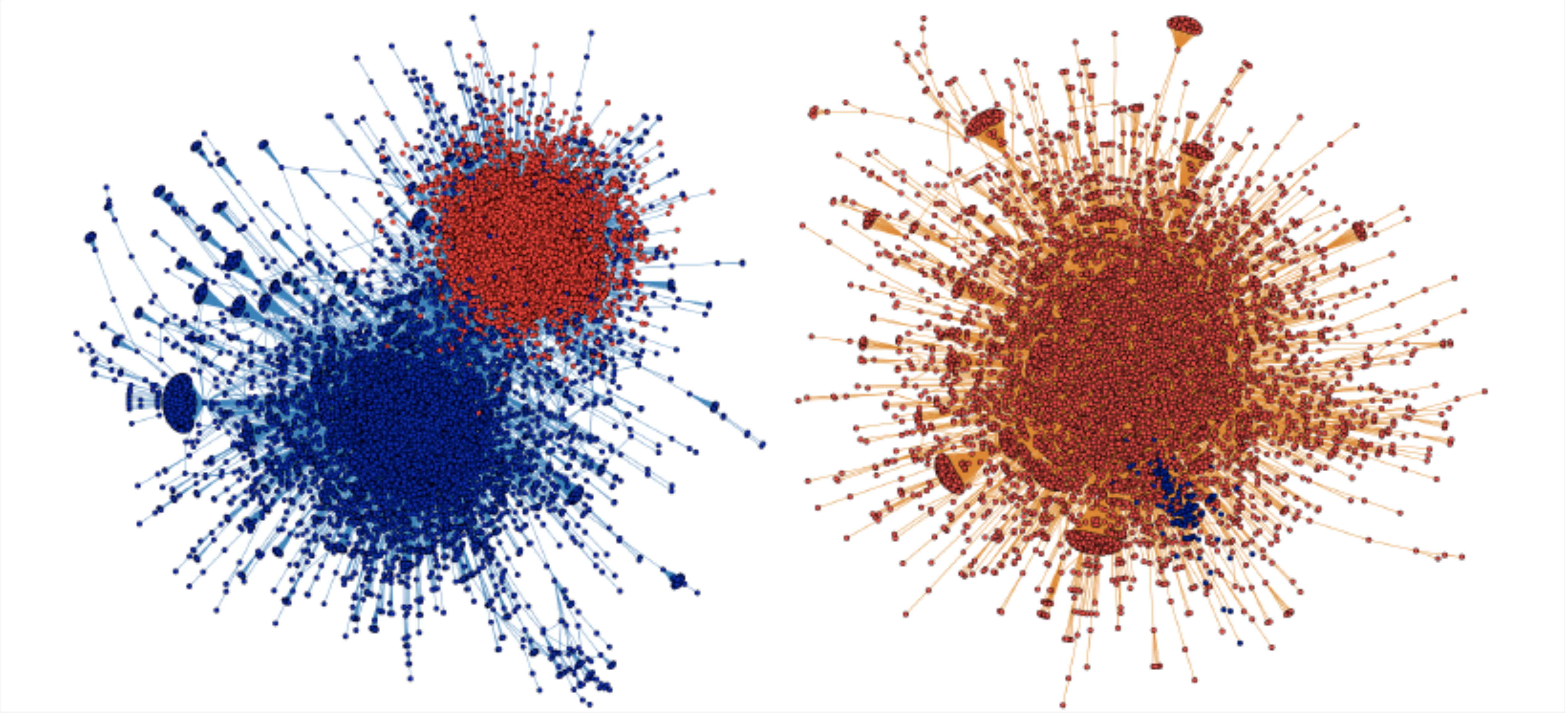


Figure 8: Time lag for blogs and news media. Thread volume in blogs reaches its peak typically 2.5 hours after the peak thread volume in the news sources. Thread volume in news sources increases slowly but decrease quickly, while in blogs the increase is rapid and decrease much slower.

@ Greene (2011)

Graphs EG2: Politics



Conover, M., Ratkiewicz, J., Francisco, M., Gonçalves, B., Menczer, F., & Flammini, A. (2011, July). Political polarization on twitter. In ICWSM.

Graphs EG2: Politics

- People with different political biases post on Twitter in subtly different ways; that may be reflected in the hahstags they use (#irishwater, #dontpayforwater)
- 250K Tweets from 2010 US Congress Election
- Political retweets highly segregated, no cross-posting between left- and right-leaning users
- Mention networks are heterogenous; based on similar hashtags, mentions and re-tweets

Conover, M., Ratkiewicz, J., Francisco, M., Gonçalves, B., Menczer, F., & Flammini, A. (2011, July). Political polarization on twitter. In ICWSM.

Clustering

Evaluation of Clusters (Cluster Validation)

Evaluation

- Are your clusters any good? How can u tell?
- **Internal:**
  - Correlation
  - Similarity
- **External:** (Jaccards, Mutual Information)
  - Rand Measure
  - Purity
  - Entropy

Internal Evaluation: Idea

The idea is that you look at the internal structure of the clusters and how they sit relative to one another using various measures: like correlations, similarity, Sums of Square Error (SSE) and so on…

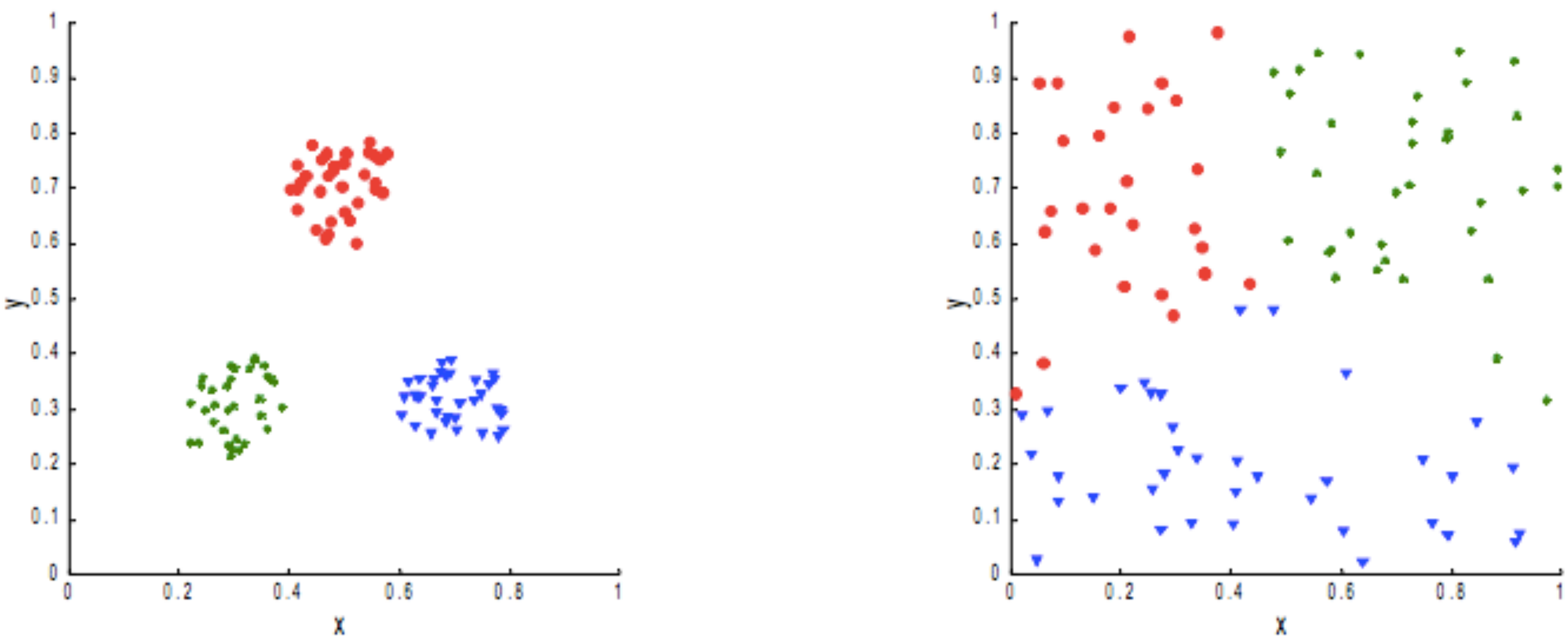
*Corr. * Davies-Bouldin Index * Dunn Index * SSE*

Internal: Correlation

- Create two matrices:
  - *Distance*: a square matrix, with *i* rows and *j* columns; where a given *i,j* cell has an entry for similarity of these two items to one another, or the distance they are apart
  - *Incidence*: an aligned matrix, same *ij* items, where (a) one row, one column for each data point; (b) entry is 1 if the associated pair of points belong to the same cluster, (c) entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between the two matrices: only n(n-1)/2 entries need to be calculated, as they are symmetric
- High correlation shows points in same cluster are close to each other

http://www.cs.kent.edu/~jin/DM08/ClusterValidation.pdf

Internal: Correlation



Corr = -0.9235 **Corr = -0.5810**

http://www.cs.kent.edu/~jin/DM08/ClusterValidation.pdf

Internal: Similarity

- Often used to compare competing algorithms; best is the one with highest within-cluster similarity & lowest between-cluster similarity

- Do tests to determine the internal similar of items in the cluster and check whether they are closer than ones in other clusters

Internal: Similarity

Davies–Bouldin index

The Davies–Bouldin index can be calculated by the following formula:

$$DB = \frac{1}{n}\sum_{i=1}^{n}\max_{j\neq i}\left(\frac{\sigma_i+\sigma_j}{d(c_i,c_j)}\right)$$

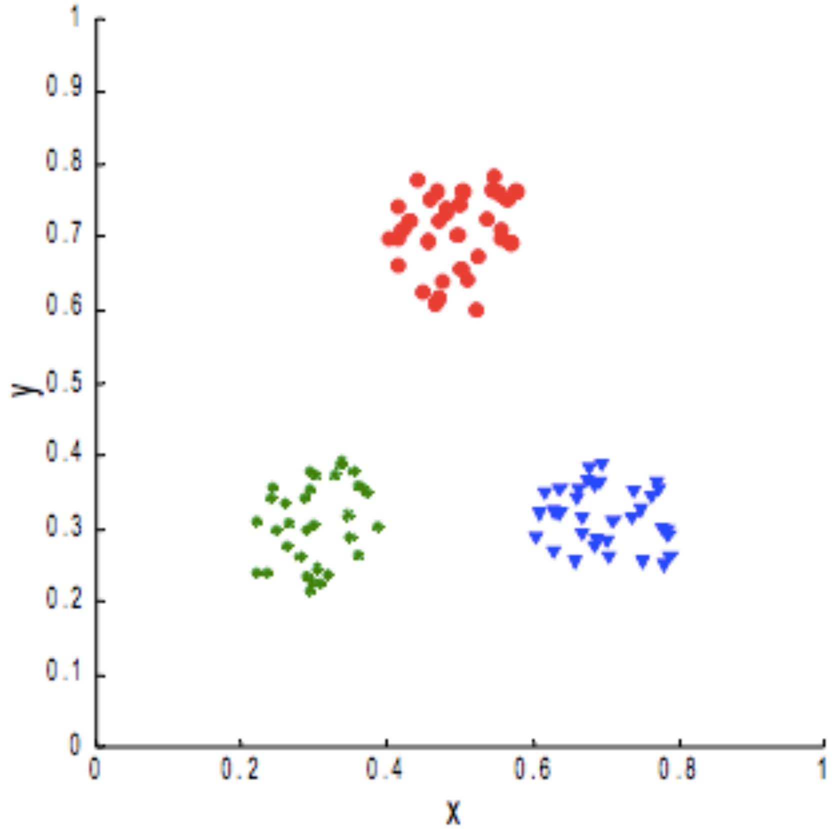
where n is the number of clusters, c_x is the centroid of cluster x, σ_x is the average distance of all elements in cluster x to centroid c_x, and $d(c_i, c_j)$ is the distance between centroids c_i and c_j. Since algorithms that produce clusters with low intra-cluster distances (high intra-cluster similarity) and high inter-cluster distances (low inter-cluster similarity) will have a low Davies–Bouldin index, the clustering algorithm that produces a collection of clusters with the smallest Davies–Bouldin index is considered the best algorithm based on this criterion.

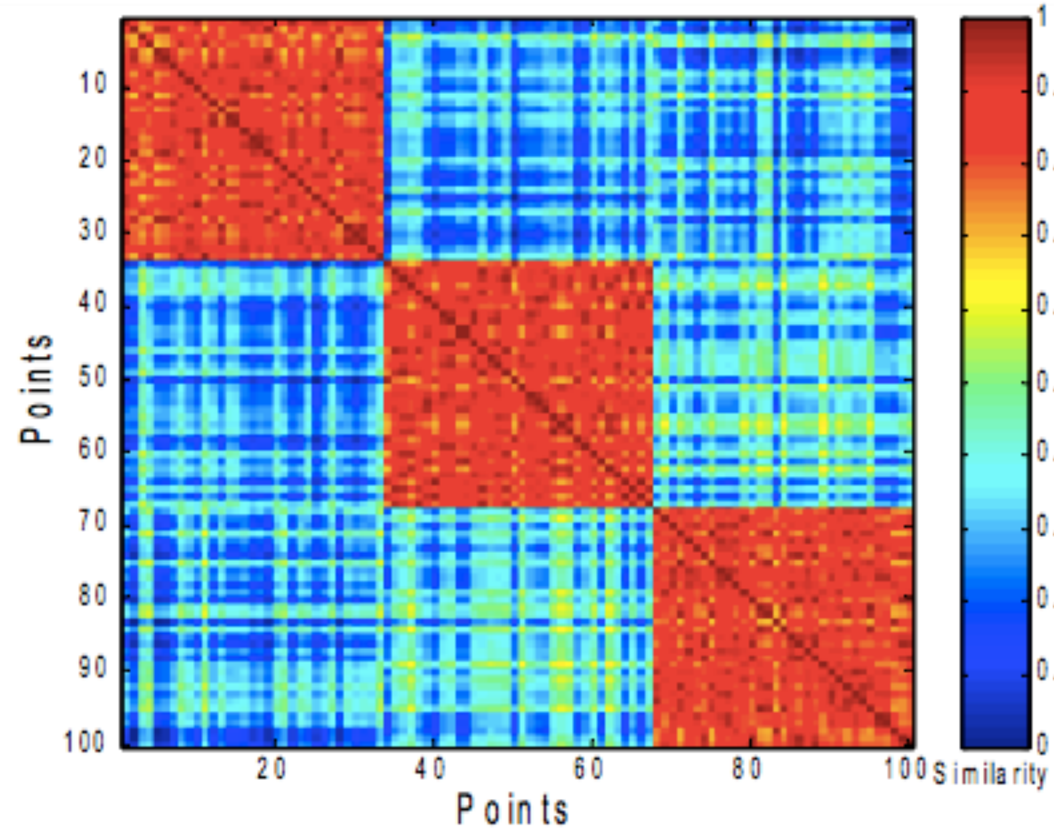
- **Dunn index**

  The Dunn index aims to identify dense and well-separated clusters. It is defined as the ratio between the minimal inter-cluster distance to maximal intra-cluster distance. For each cluster partition, the Dunn index can be calculated by the following formula:[31]

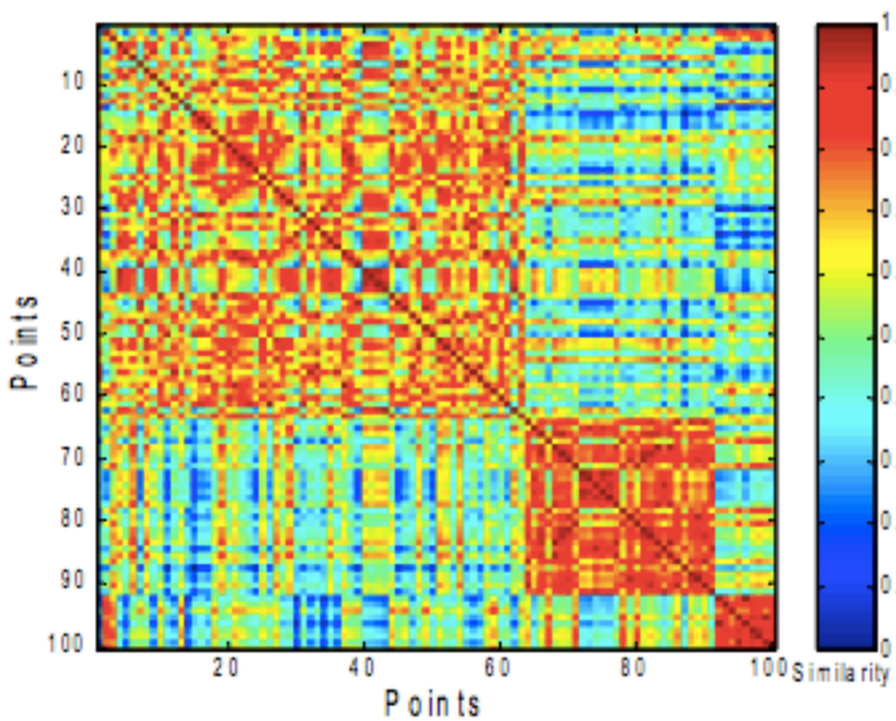
  $$D = \frac{\min_{1\leq i<j\leq n} d(i,j)}{\max_{1\leq k\leq n} d'(k)}\,,$$

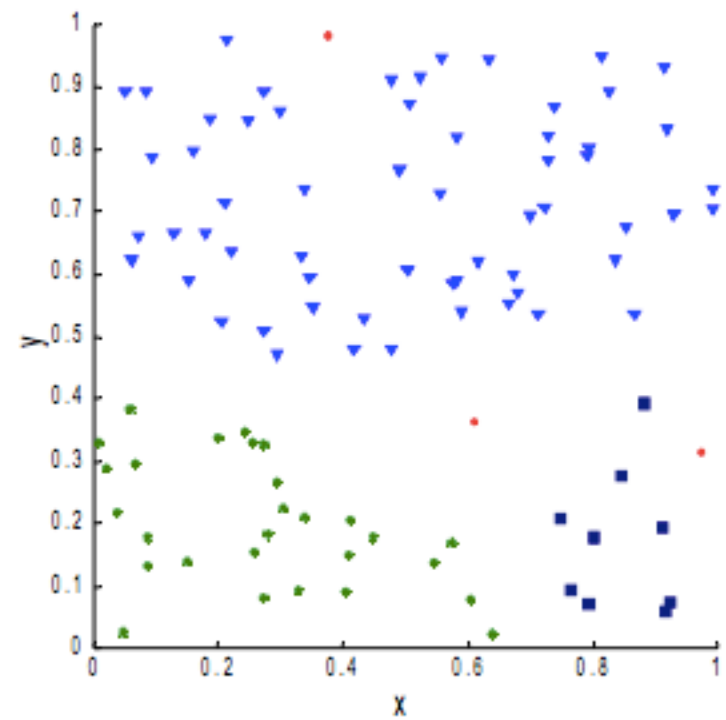
  where *d*(*i*,*j*) represents the distance between clusters *i* and *j*, and *d* '(*k*) measures the intra-cluster distance of cluster *k*. The inter-cluster distance *d*(*i*,*j*) between two clusters may be any number of distance measures, such as the distance between the centroids of the clusters. Similarly, the intra-cluster distance *d* '(*k*) may be measured in a variety ways, such as the maximal distance between any pair of elements in cluster *k*. Since internal criterion seek clusters with high intra-cluster similarity and low inter-cluster similarity, algorithms that produce clusters with high Dunn index are more desirable.



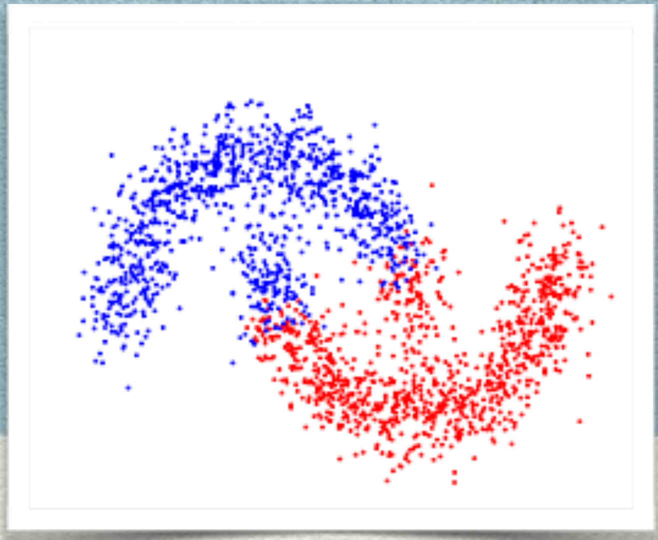


- Clusters in random data are not so crisp





Internal: Some Points



- Note, these internal methods are never conclusive; eg., the fact that k-means can be screwed by convex clusters, will not be found
- Note, many methods build in the evaluation metric, so it gets circular
- Probably, best for inter-algorithm comparison

External Evaluation: Idea

Idea is you go outside to some external source of "right" answers; then its a comparison between what this gold standard says and what we have found: either wrt individual items or the class (item is *x*, or item is in class-*y* not class-*x*)

*Rand * Jaccard * F-measure * Purity * Entropy*

External: Rand & Jaccard

Rand measure (William M. Rand 1971)[33]

The Rand index computes how similar the clusters (returned by the clustering algorithm) are to the benchmark classifications. One can also view the Rand index as a measure of the percentage of correct decisions made by the algorithm. It can be computed using the following formula:

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. One issue with the Rand index is that false positives and false negatives are equally weighted. This may be an undesirable characteristic for some clustering applications. The F-measure addresses this concern, as does the chance-corrected adjusted Rand index.

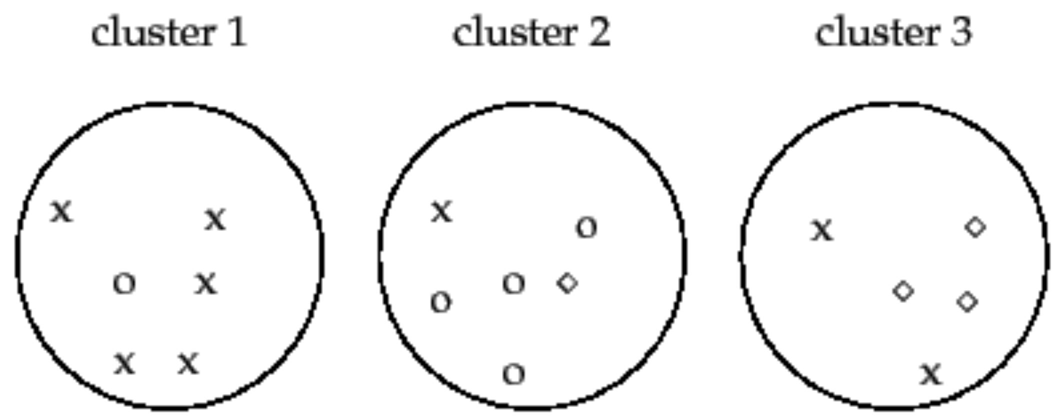
Jaccard index

The Jaccard index is used to quantify the similarity between two datasets. The Jaccard index takes on a value between 0 and 1. A index of 1 means that the two dataset are identical, and an index of 0 indicates that the datasets have no common elements. The Jaccard index is defined by the following formula:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN}$$

This is simply the number of unique elements common to both sets divided by the total number of unique elements in both sets.

External: Purity



► **Figure 16.1** Purity as an external evaluation criterion for cluster quality. Majority class and number of members of the majority class for the three clusters are: x, 5 (cluster 1); o, 4 (cluster 2); and ⋄, 3 (cluster 3). Purity is $(1/17) \times (5+4+3) \approx 0.71$.

To compute *purity* , each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned documents and dividing by N. Formally:

$$\text{purity}(\Omega, \mathbb{C}) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j| \quad (182)$$

where $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ is the set of clusters and $\mathbb{C} = \{c_1, c_2, \ldots, c_J\}$ is the set of classes. We interpret ω_k as the set of documents in ω_k and c_j as the set of documents in c_j in Equation 182.

We present an example of how to compute purity in Figure 16.4 . Bad clusterings have purity values close to 0, a perfect clustering has a purity of 1 . Purity is compared with the other three measures discussed in this chapter in Table 16.2 .

http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html

External: Entropy

- Or, you can use entropy; if the items in a cluster are “the same” then entropy is low, if they are “different” then entropy is high

Clustering

Conclusions

Unsupervised: Clustering

- K-Means
- Hierarchical Clustering
- Graph-Based Clustering

Selecting Some…

- **Supervised Learning: Classification**
  - *k* Nearest Neighbours
  - Naive Bayes
  - Logistic Regression
  - Support Vector Machine
- **Unsupervised Learning: Clustering** (next lecture)
  - K-Means
  - Hierarchical Clustering
  - Graph-based Clustering

http://www.slideshare.net/mirjalil/clustering-on-database-systems-rkm-42588852

End Bits

Unsupervised: Clustering

- K-Means
- Hierarchical Clustering
- Graph-Based Clustering
- Expectation Maximisation
- (Blind Separation & Latent)