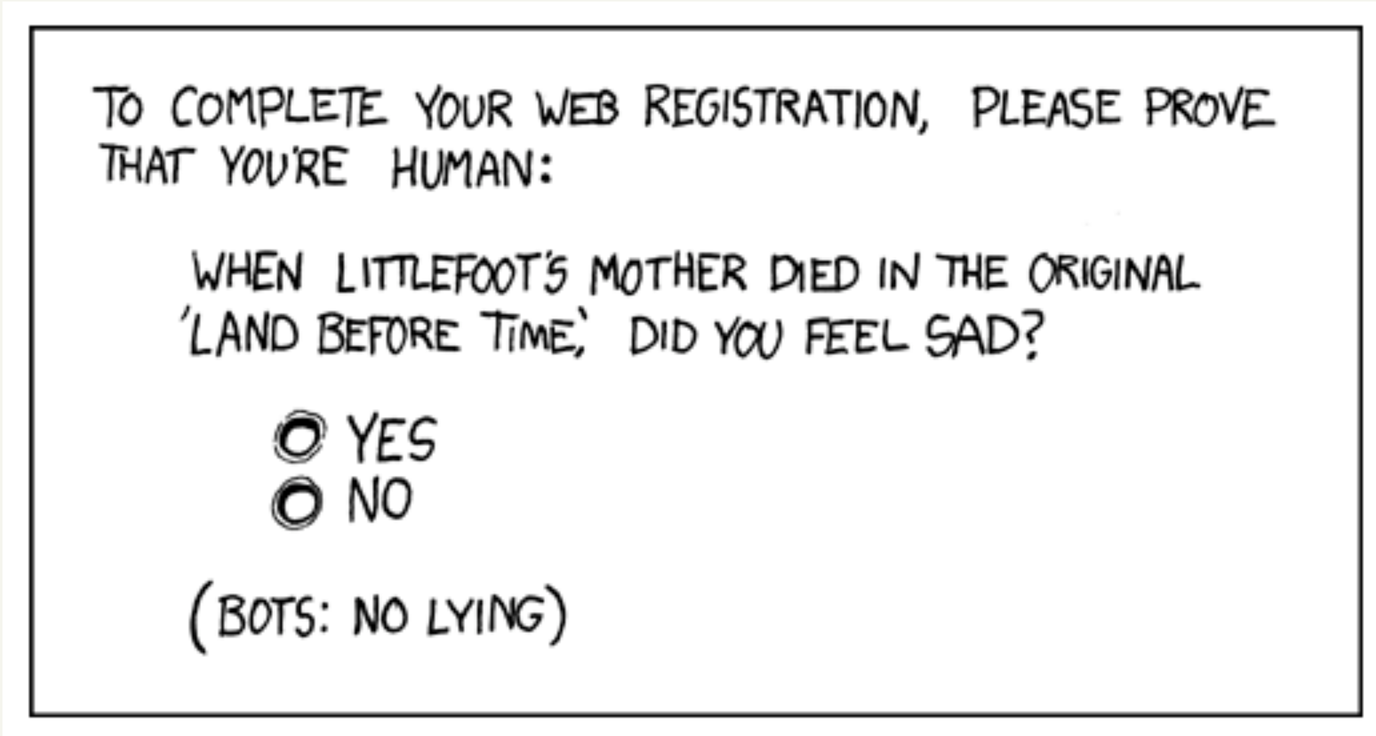


http://www.xkcd.com/233/

Text Clustering

David Kauchak
cs160
Fall 2009

adapted from:
http://www.stanford.edu/class/cs276/handouts/lecture17-clustering.ppt

Administrative

- 2nd status reports
- Paper review

IR code changes

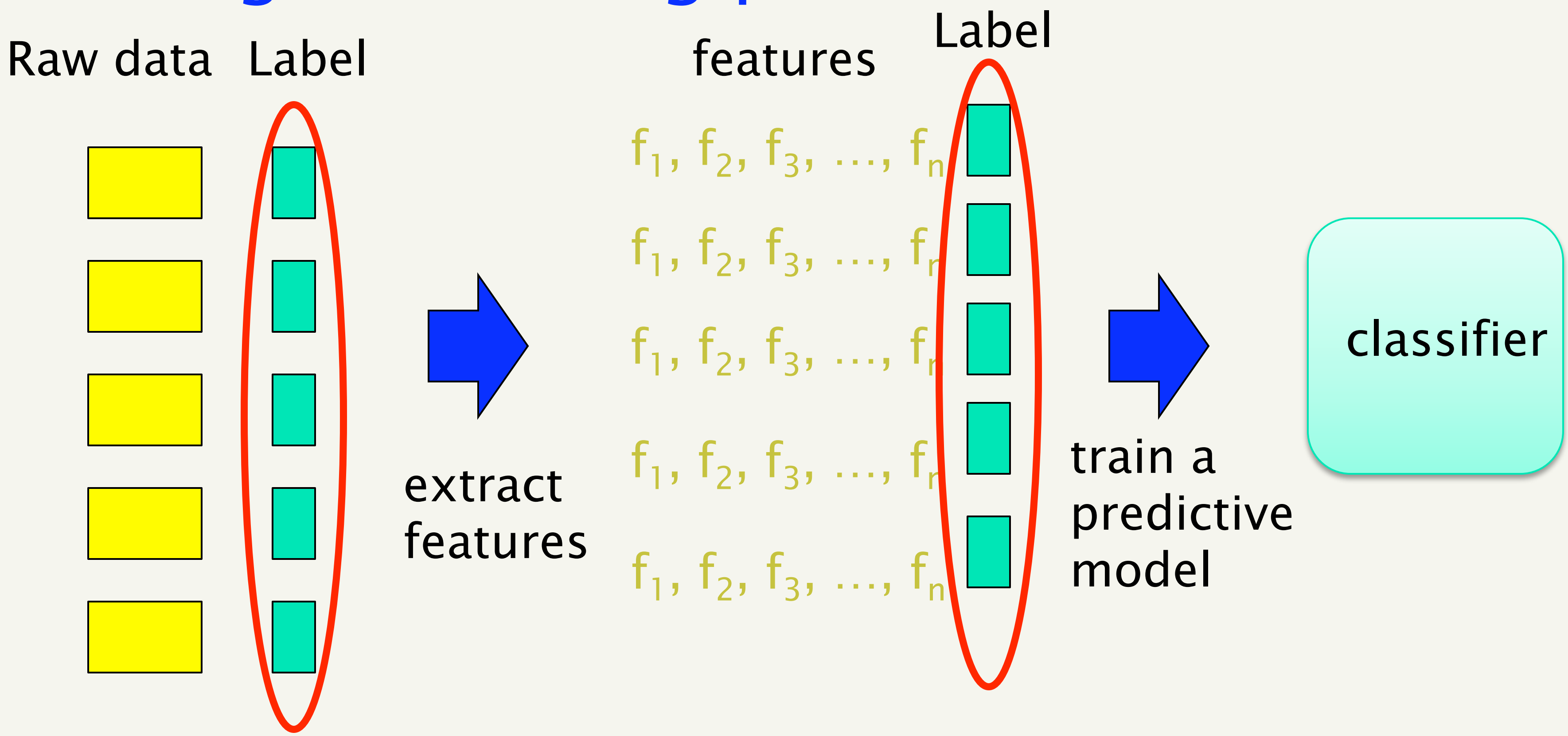
- /common/cs/cs160/project/
  - requires the mysql…jar
  - will get into svn, but for now can look at code
- TDT data set with title and paragraph breaks
  - tdt.text.title or tdt.text.p.title in main data dir
- TDT is in database
  - you can access it from any where in the lab
  - can add other data sets, but let me know
    - need to write a DocumentReader class for it
    - image datasets?
    - page rank dataset?
- http://saras.cs.pomona.edu/bursti/doc.php?id=10

IR code changes

- broke up indexing and querying steps
  - pass in an index file, instead of a set of documents
  - /common/cs/cs160/project/data has one index
- Document class
  - added title and url fields
  - added paragraph boundaries (only works for docs in database right now)
- Added DB interfacing classes
  - DBDocReader (a DocReader) for iterating through documents in the database
  - DBDocFetcher to fetch documents based on document id for query time document fetching

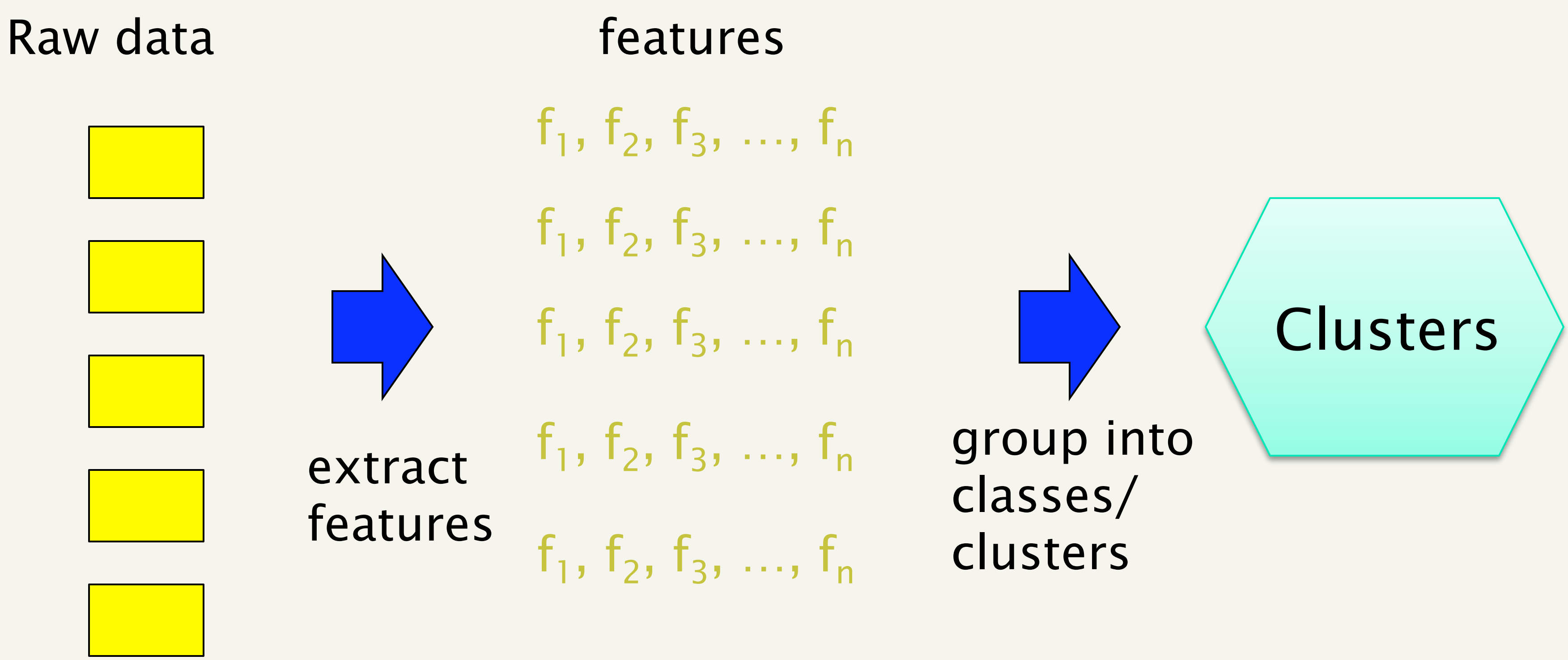
Supervised learning

Training or learning phase



User "supervision", we're given the labels (classes)

Unsupervised learning



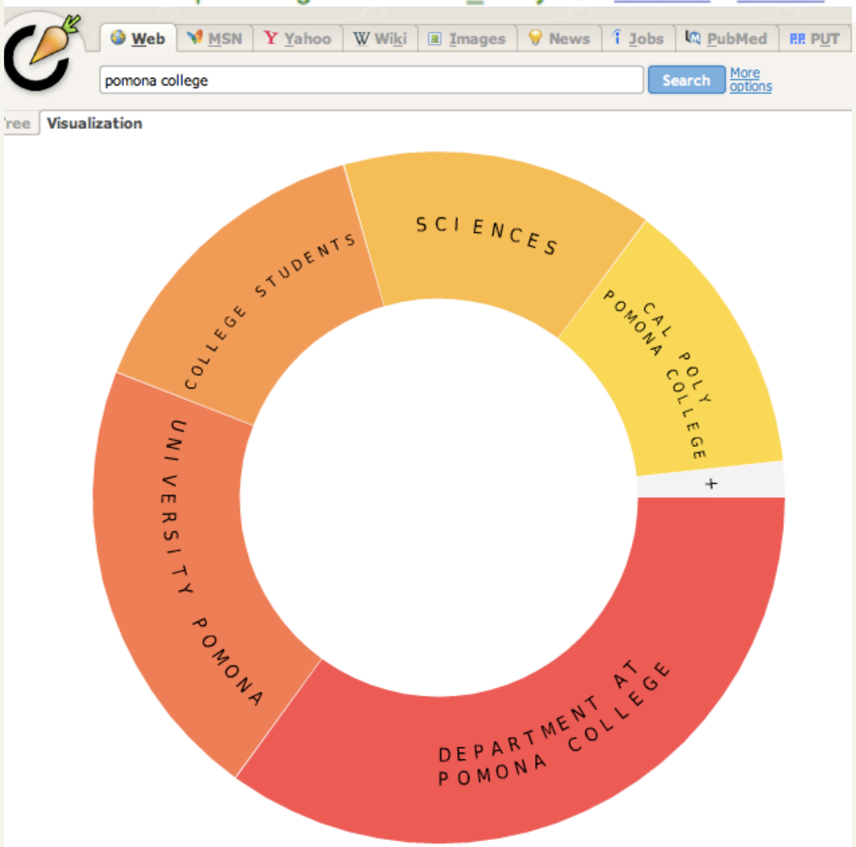
No "supervision", we're only given data and want to find natural groupings

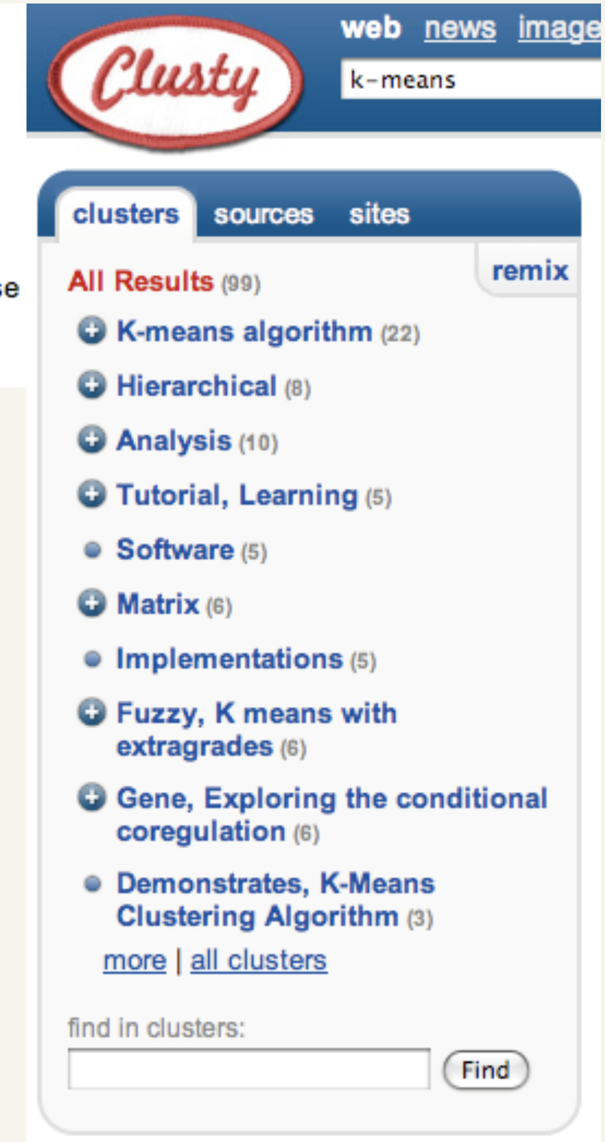
What is clustering?

- Clustering: the process of grouping a set of objects into classes of similar objects
  - Documents within a cluster should be similar
  - Documents from different clusters should be dissimilar

- How might this be useful for IR?

Applications of clustering in IR





group related docs

http://search.carrot2.org/

For improving search recall

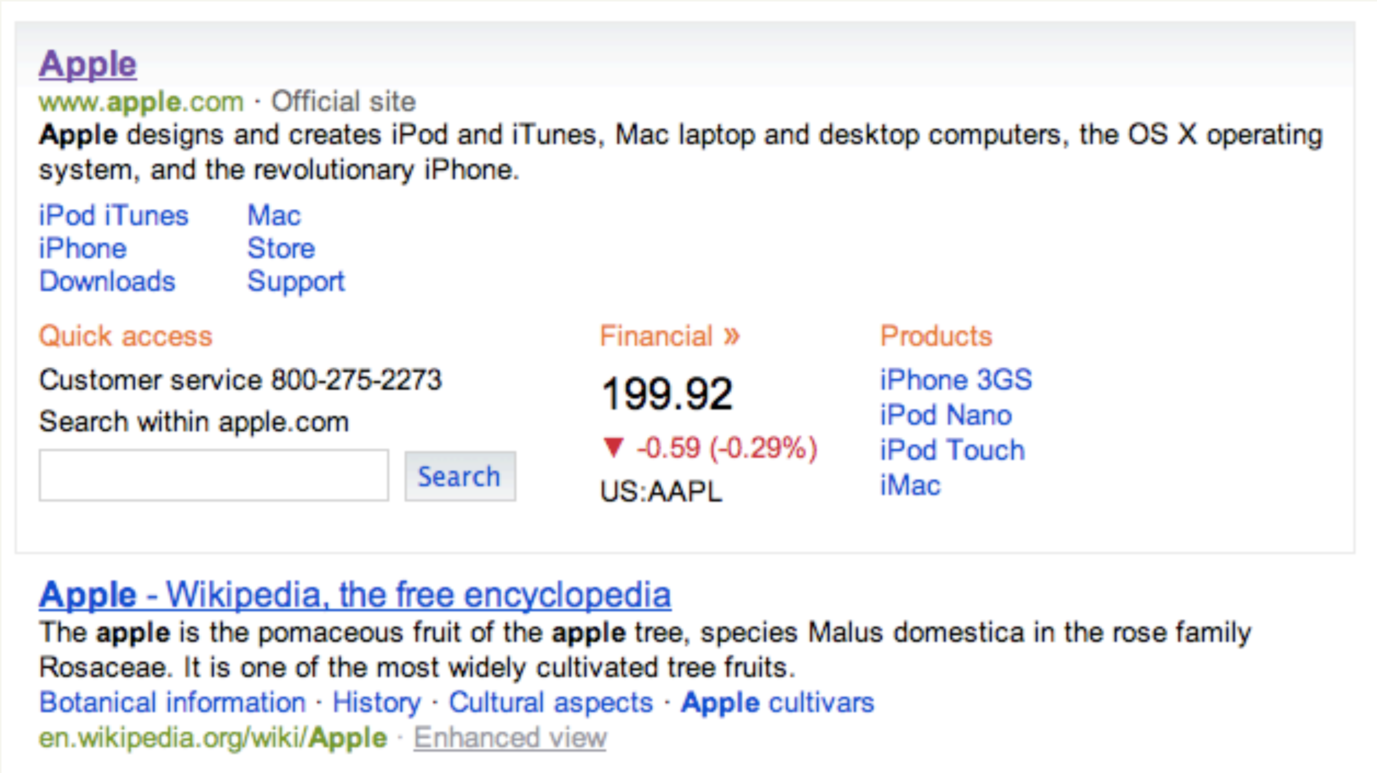
- *Cluster hypothesis* - Documents in the same cluster behave similarly with respect to relevance to information needs

- To improve search recall:
  - Cluster docs in corpus
  - When a query matches a doc *D*, also return other docs in the cluster containing *D*

- Hope if we do this: The query “car” will also return docs containing *automobile*

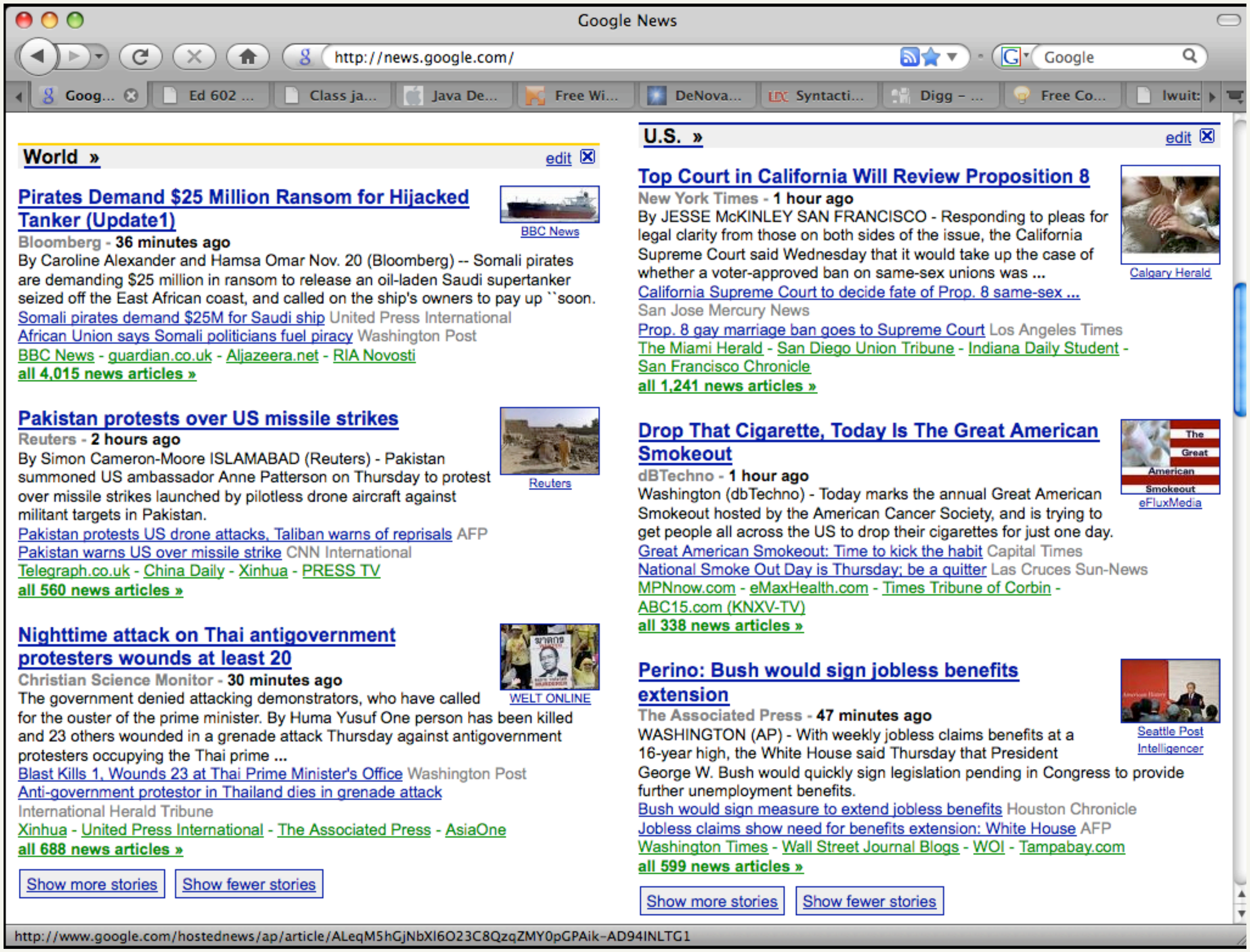
How is this different from the previous slide?

Applications of clustering in IR



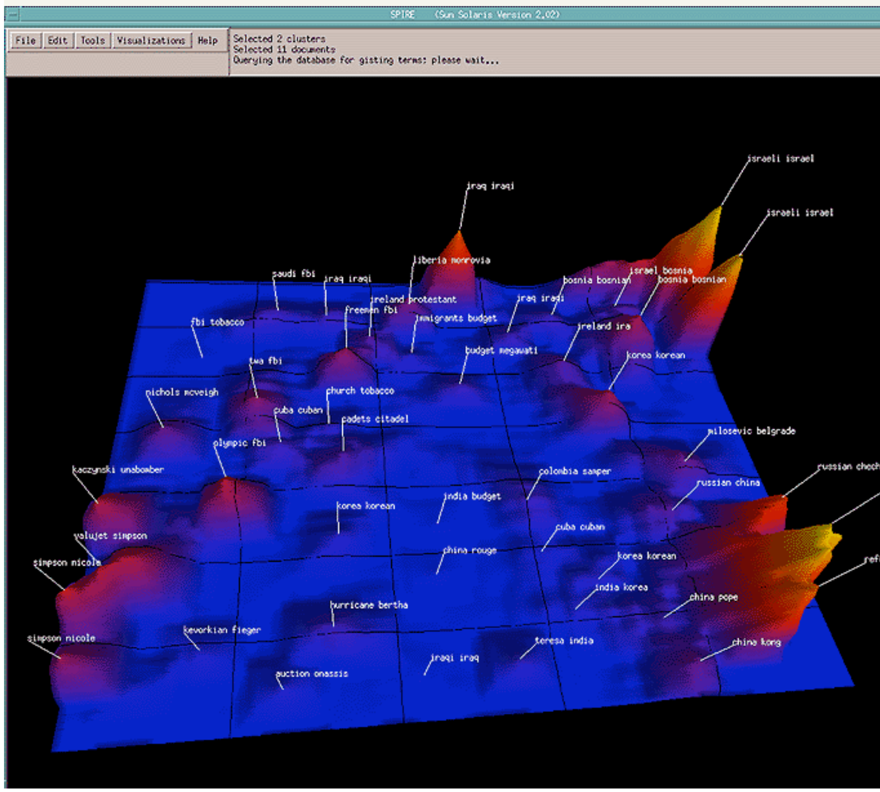
vary search results over clusters/ topics to improve recall

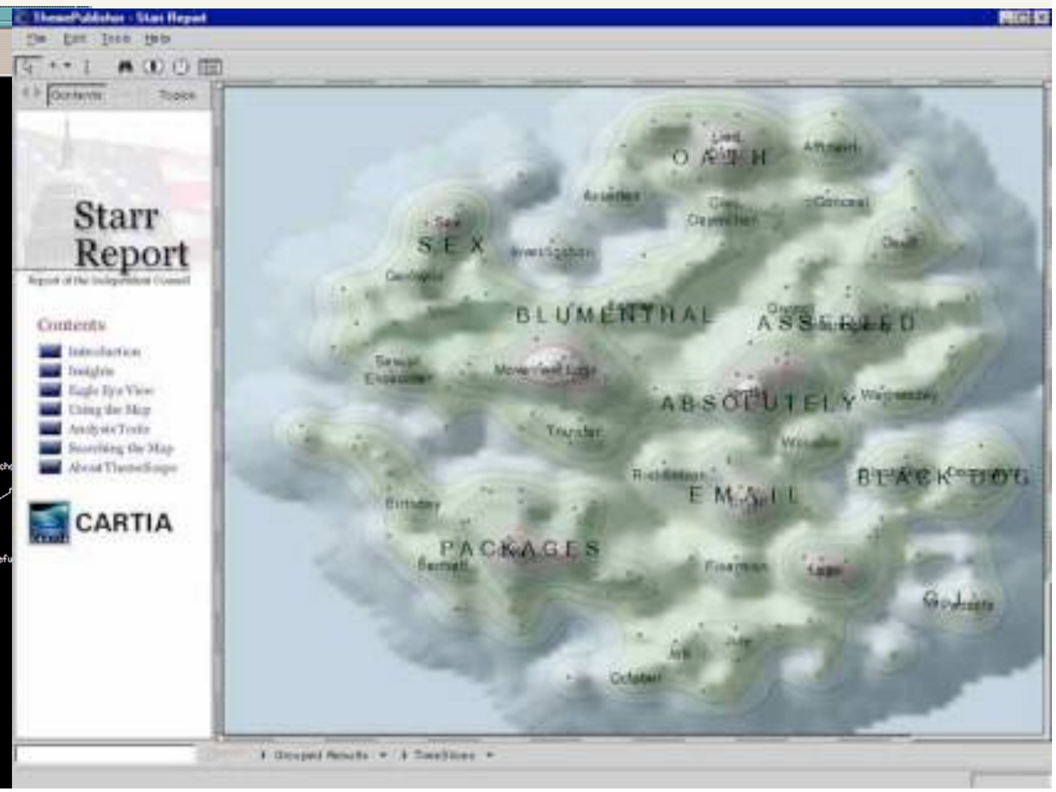
Google News: automatic clustering gives an effective news presentation metaphor



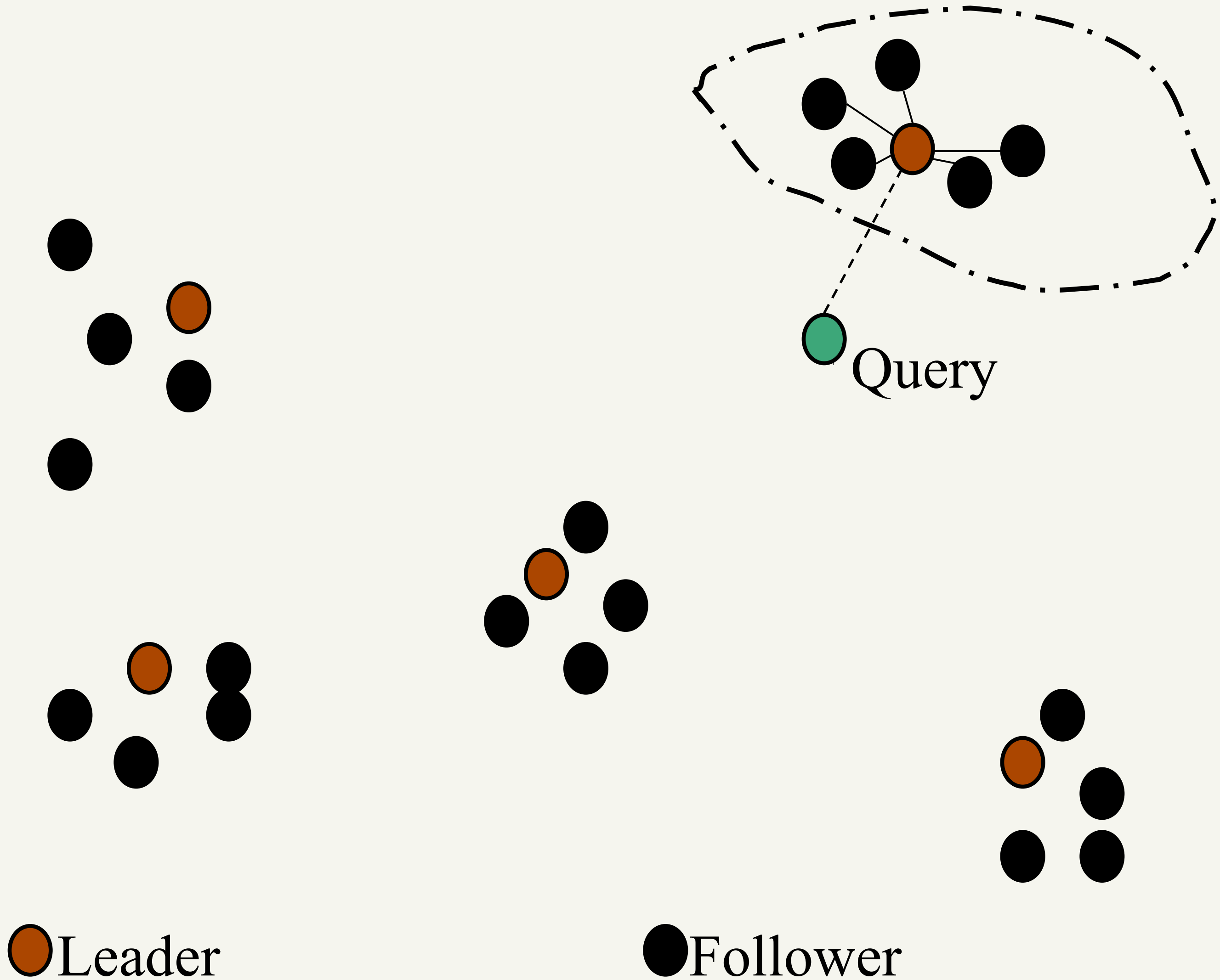
For visualizing a document collection and its themes

- Wise et al, “Visualizing the non-visual” PNNL
- ThemeScapes, Cartia
  - [Mountain height = cluster size]

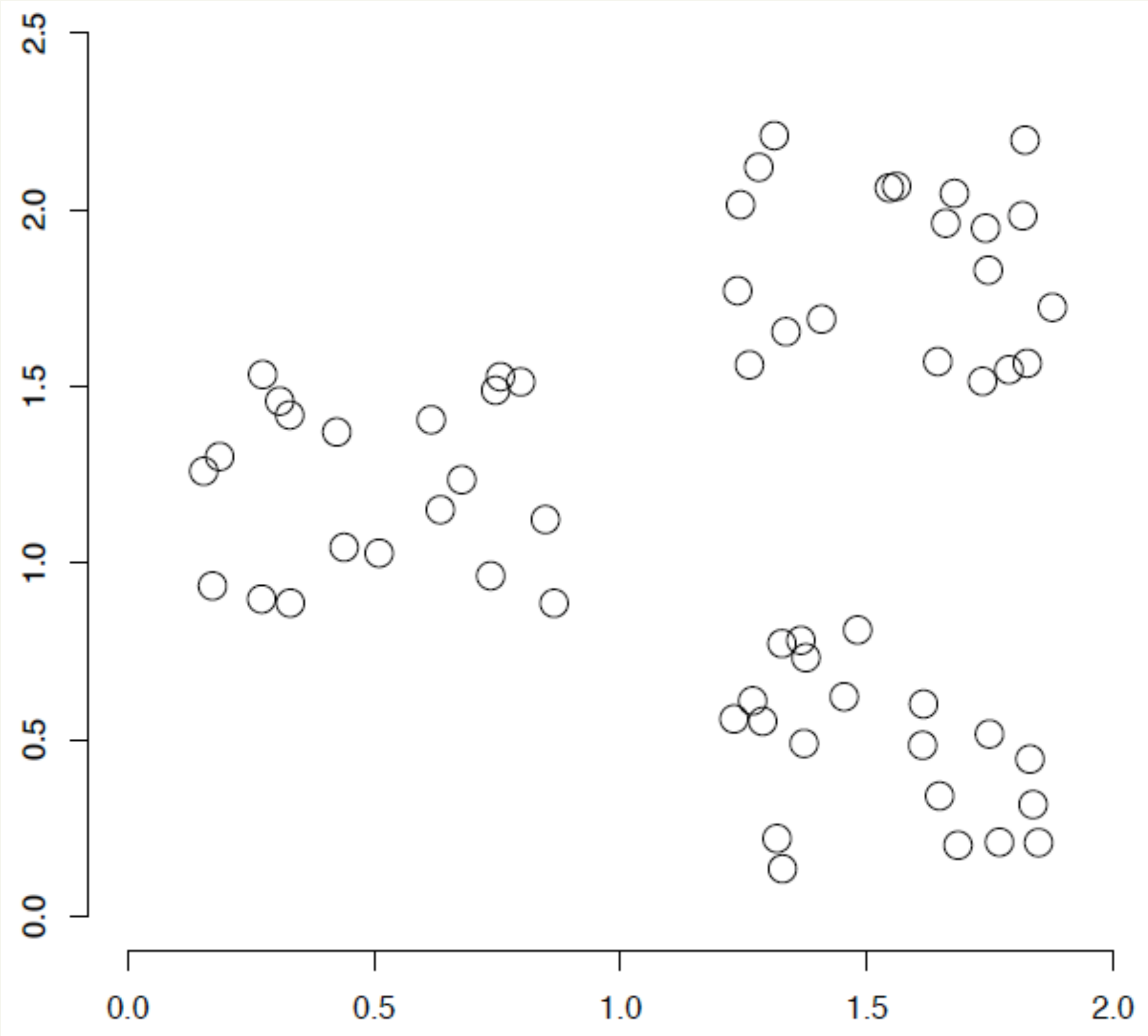




Faster vectors space search: cluster pruning



A data set with clear cluster structure



What are some of the issues for clustering?

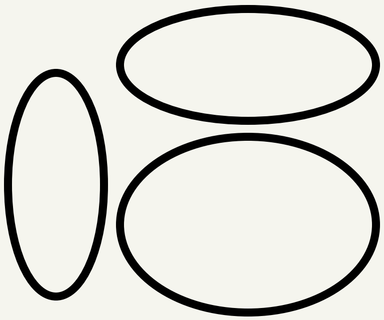
What clustering algorithms have you seen/ used?

Issues for clustering

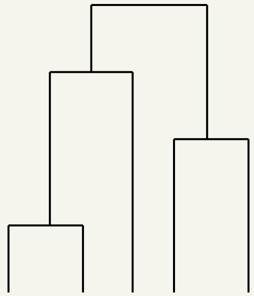
- Representation for clustering
  - Document representation
    - Vector space? Normalization?
  - Need a notion of similarity/distance
- Flat clustering or hierarchical
- Number of clusters
  - Fixed a priori
  - Data driven?

Clustering Algorithms

- Flat algorithms
  - Usually start with a random (partial) partitioning
  - Refine it iteratively
    - *K* means clustering
    - Model based clustering
  - Spectral clustering



- Hierarchical algorithms
  - Bottom-up, agglomerative
  - Top-down, divisive



Hard vs. soft clustering

- Hard clustering: Each document belongs to exactly one cluster

- Soft clustering: A document can belong to more than one cluster (probabilistic)
  - Makes more sense for applications like creating browsable hierarchies
  - You may want to put a pair of sneakers in two clusters: (i) sports apparel and (ii) shoes

Partitioning Algorithms

- Given:
  - a set of documents
  - the number of clusters *K*
- Find: a partition of *K* clusters that optimizes a partitioning criterion
  - Globally optimal: exhaustively enumerate all partitions
  - Effective heuristic methods: *K*-means and *K*-medoids algorithms

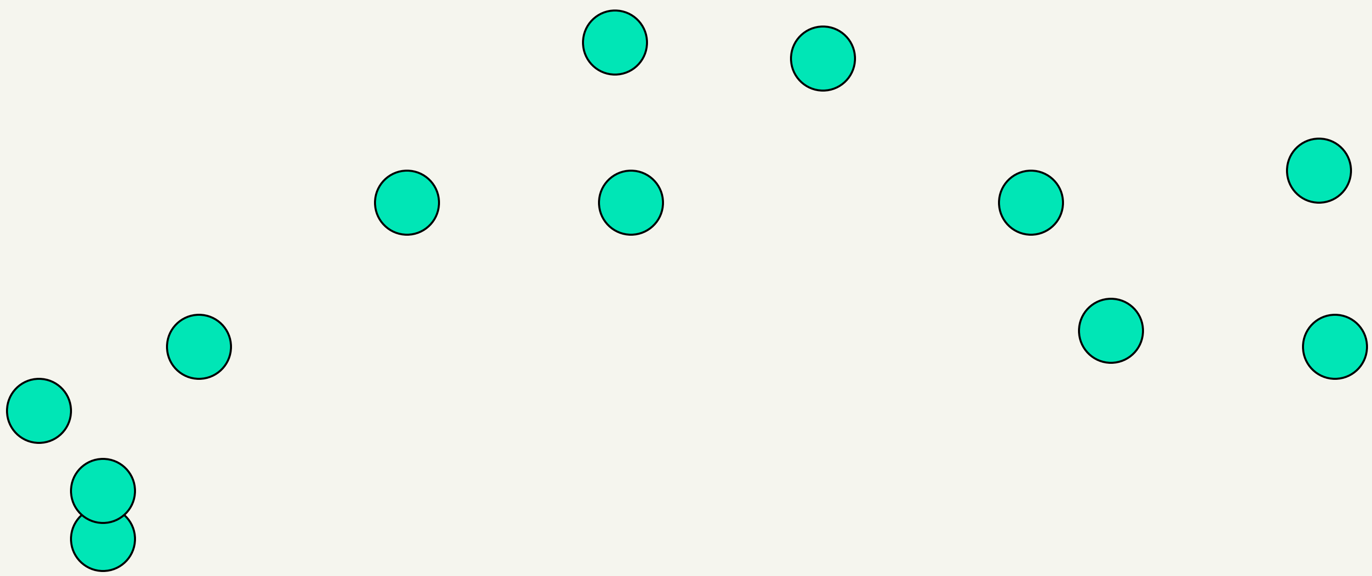
K-Means

- Start with some initial cluster centers randomly chosen as documents/points from the data

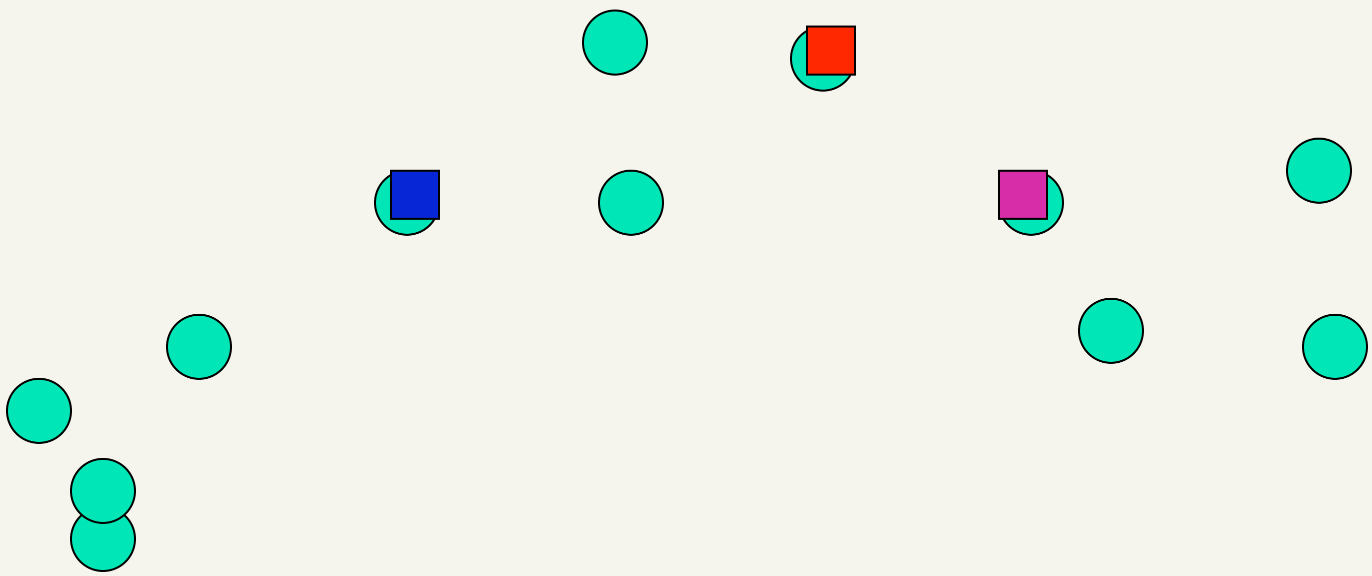
- Iterate:
  - Assign/cluster each document to closest center
  - Recalculate centers as the mean of the points in a cluster, *c*:

$$\vec{\mu}(c) = \frac{1}{|c|} \sum_{\vec{x} \in c} \vec{x}$$

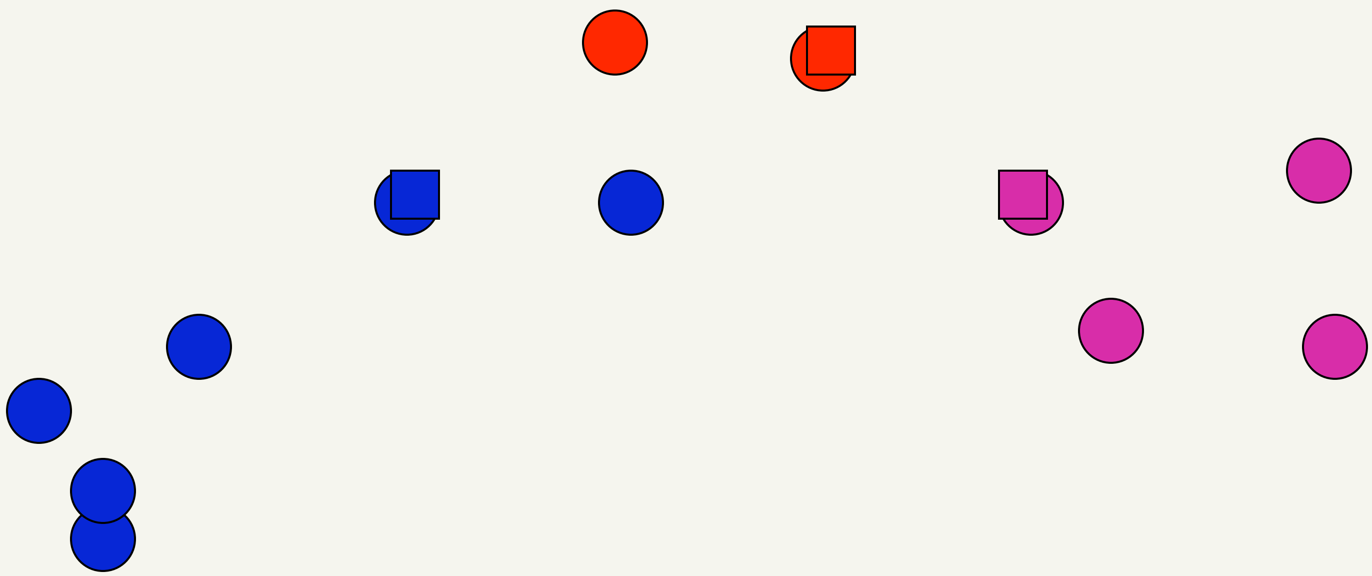
K-means: an example



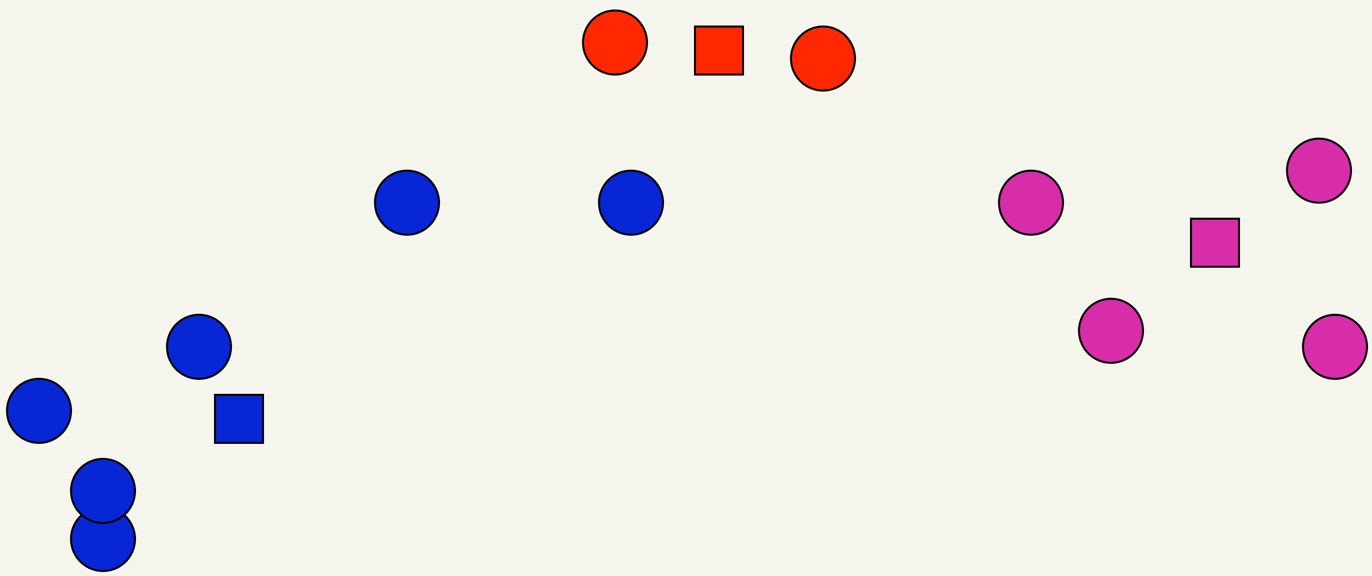
K-means: Initialize centers randomly



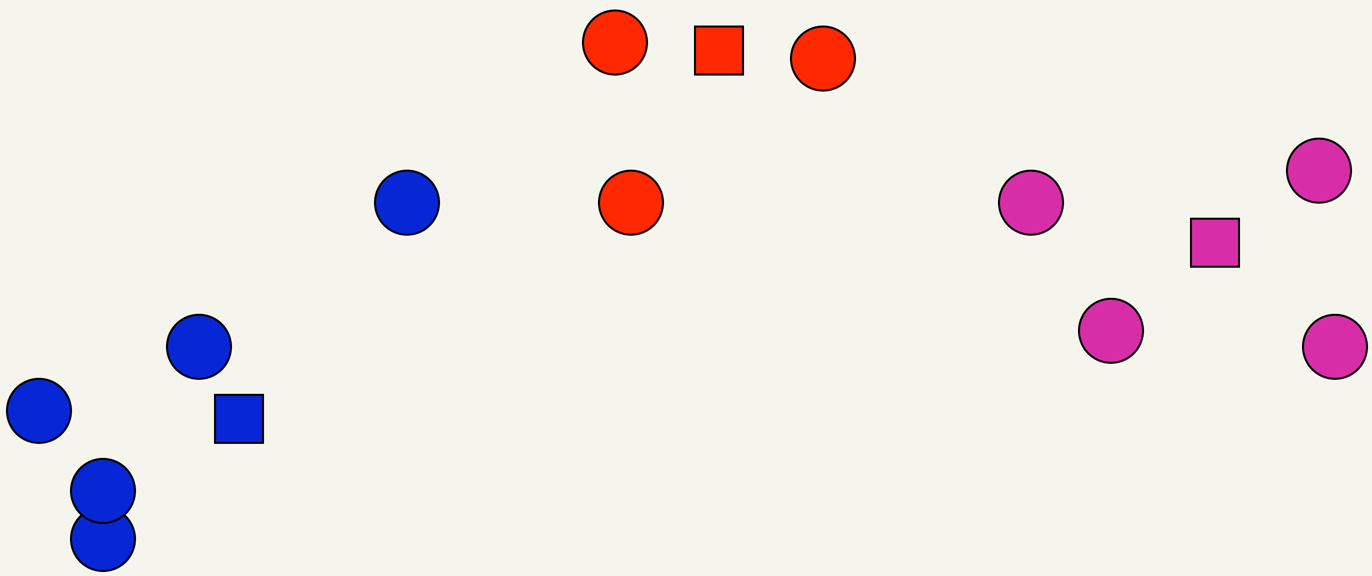
K-means: assign points to nearest center



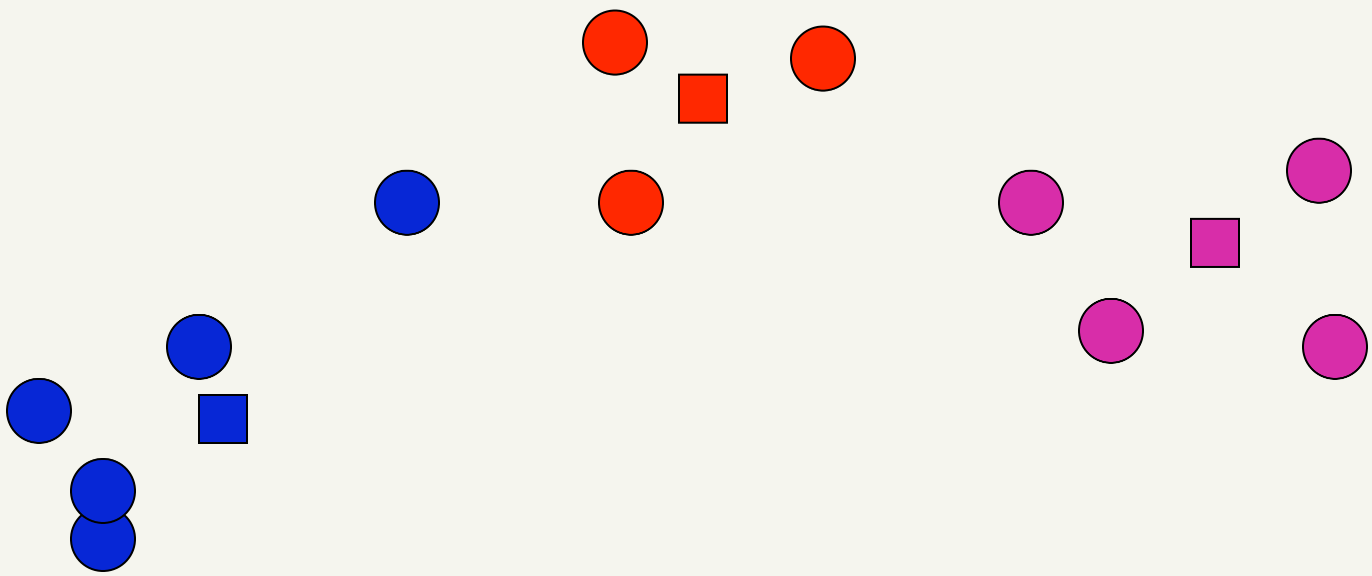
K-means: readjust centers



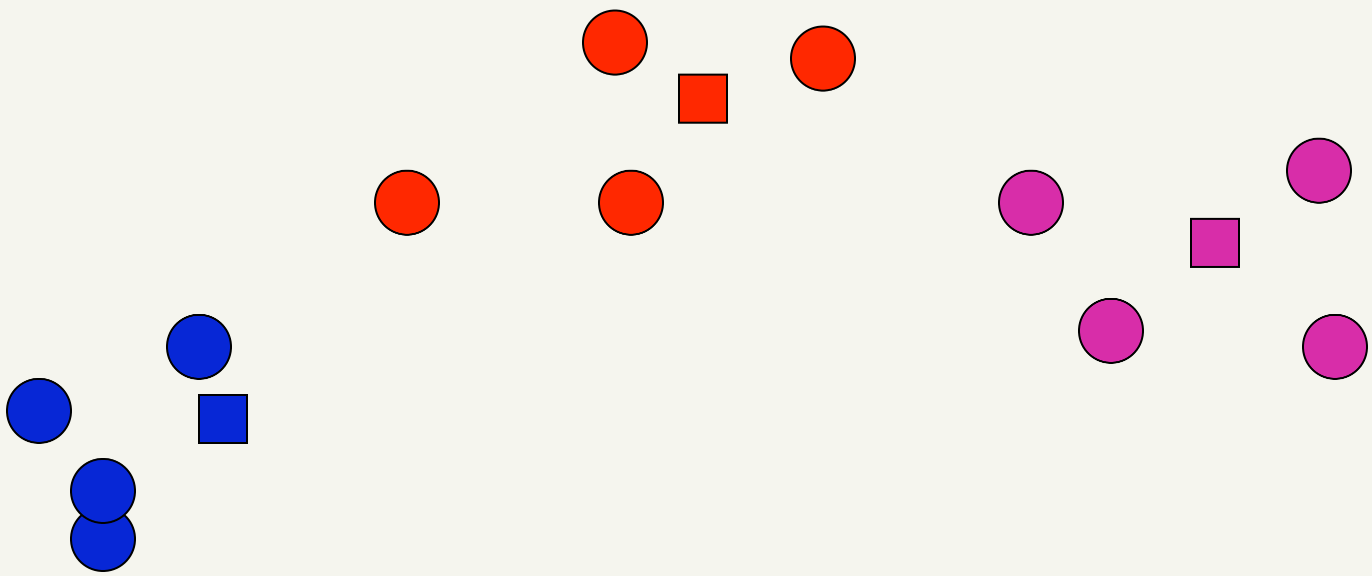
K-means: assign points to nearest center



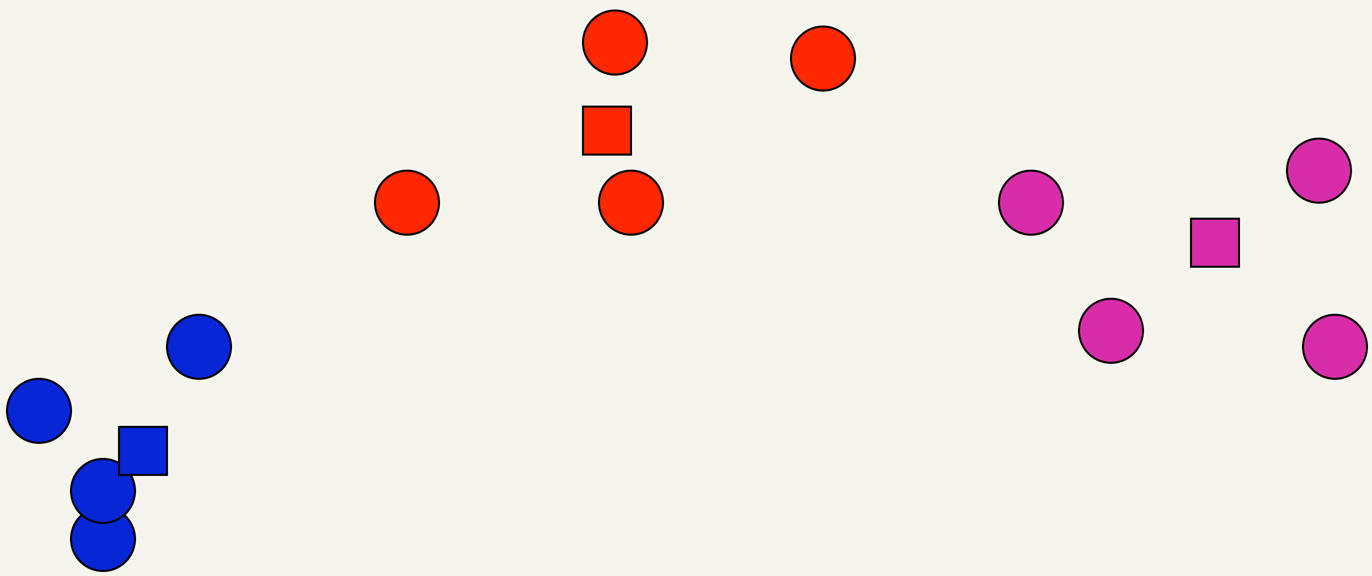
K-means: readjust centers



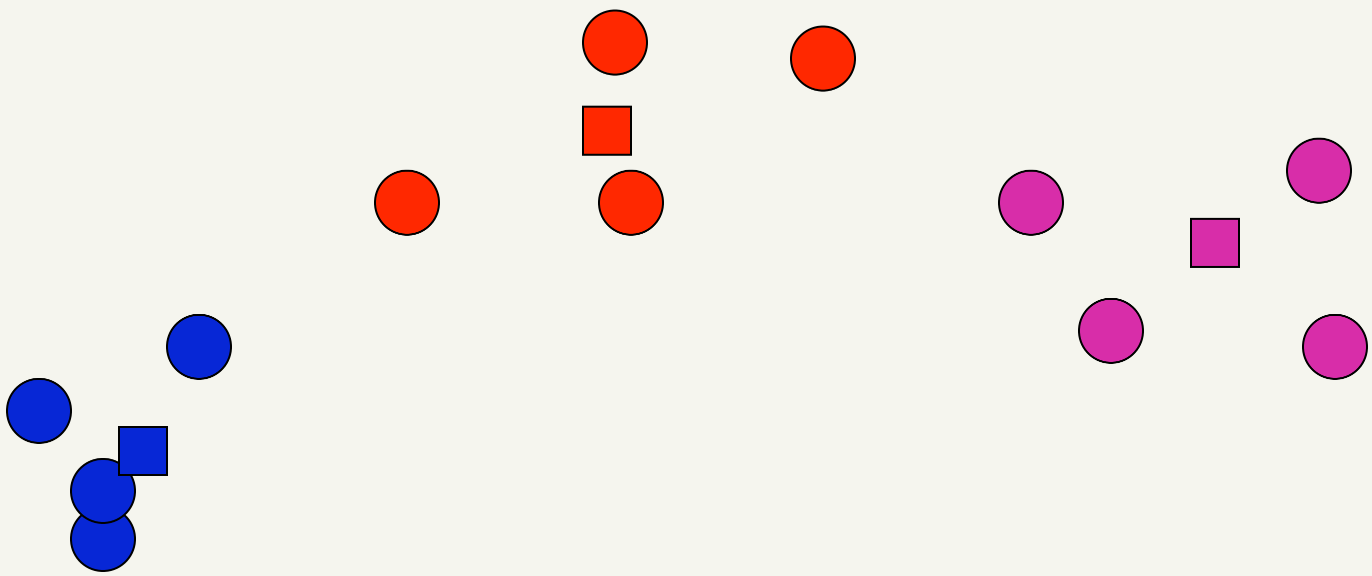
K-means: assign points to nearest center



K-means: readjust centers



K-means: assign points to nearest center

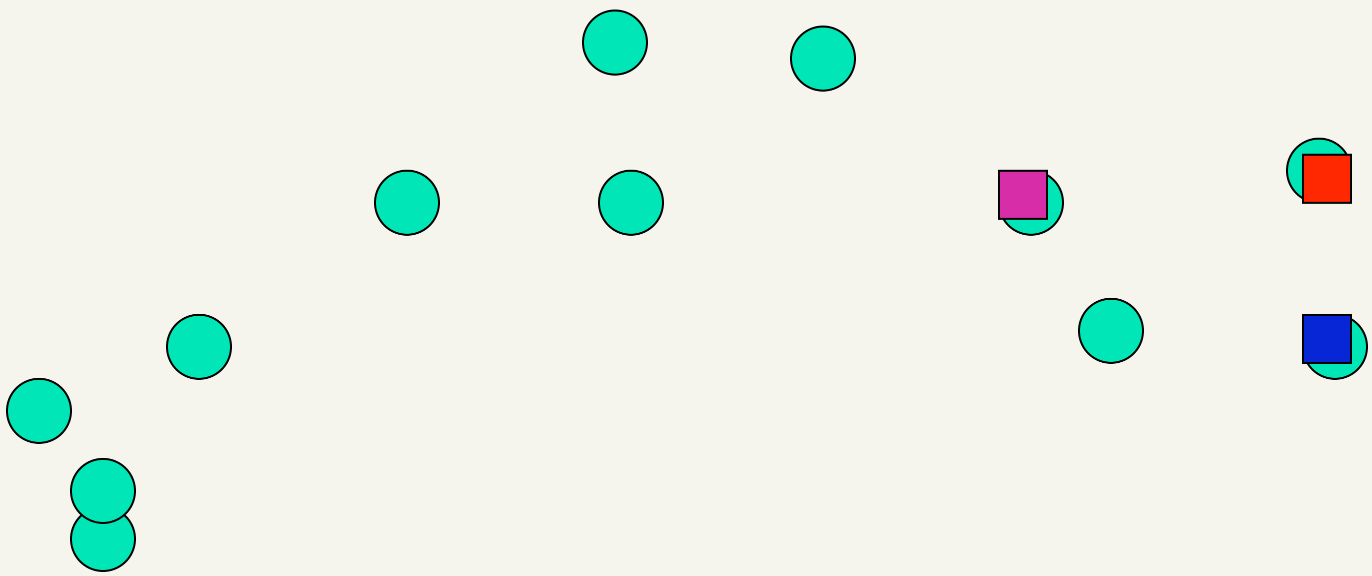


No changes: Done

K-means variations/parameters

- Initial (seed) centroids
- Convergence
  - A fixed number of iterations
  - Doc partition unchanged
  - Cluster centers don’t change
- K

K-means: Initialize centers randomly



What would happen here?

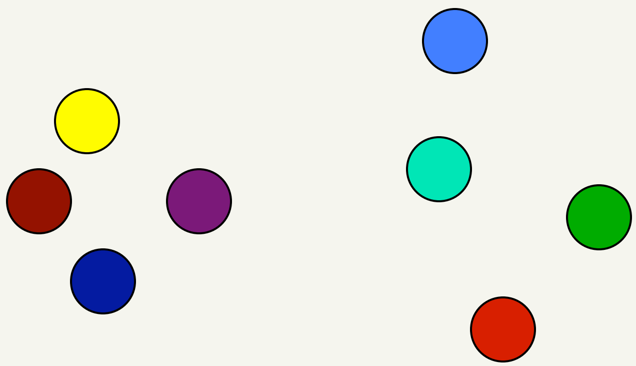
Seed selection ideas?

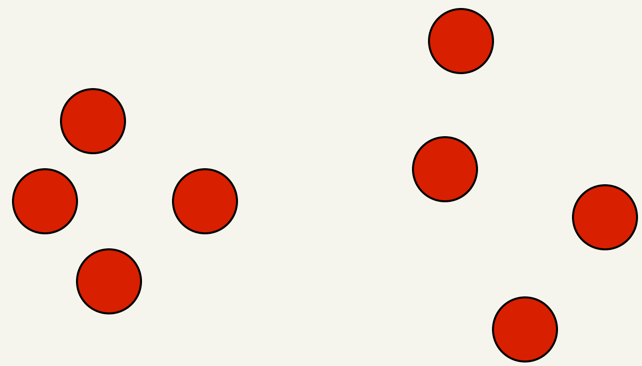
Seed Choice

- Results can vary drastically based on random seed selection
- Some seeds can result in poor convergence rate, or convergence to sub-optimal clusterings
- Common heuristics
  - Random points in the space
  - Random documents
  - Doc least similar to any existing mean
  - Try out multiple starting points
  - Initialize with the results of another clustering method

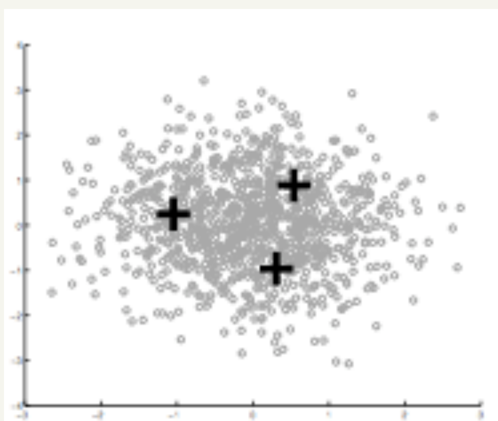
How Many Clusters?

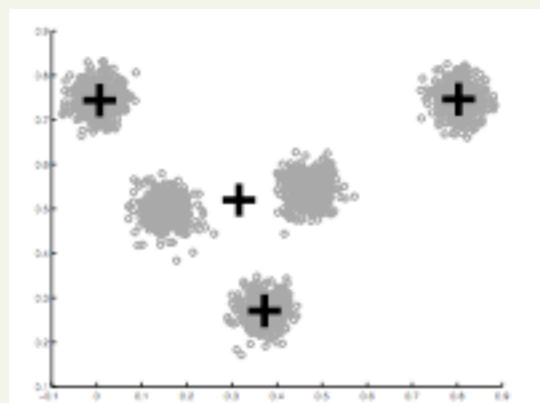
- Number of clusters *K* must be provided
- Somewhat application dependent
- How should we determine the number of clusters?





too many



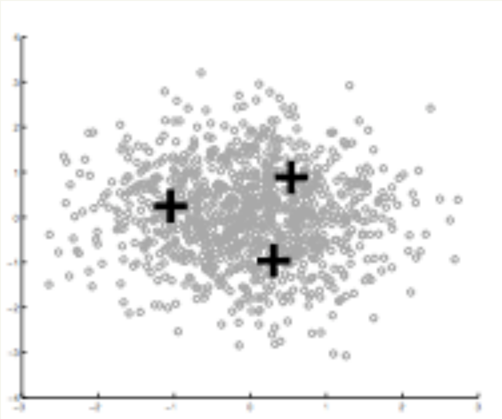


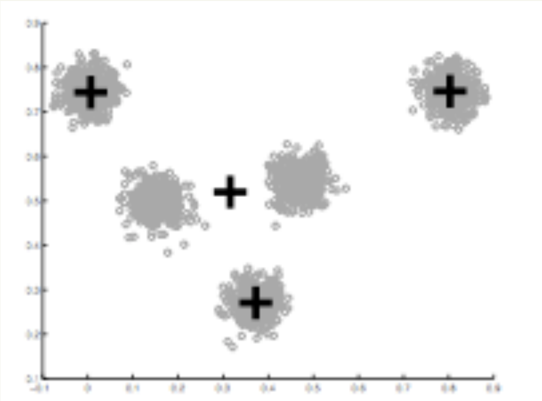
too few

One approach

- Assume data should be Gaussian (i.e. spherical)

- Test for this
  - Testing in high dimensions doesn't work well
  - Testing in lower dimensions does work well

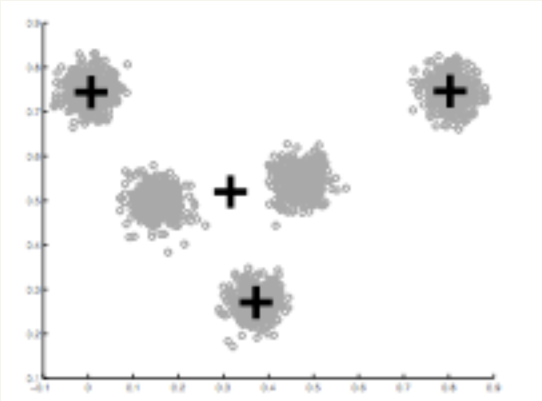




ideas?

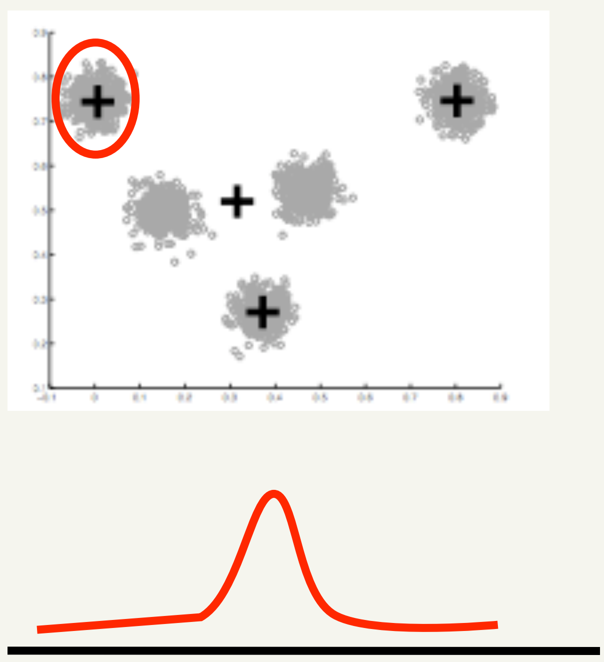
Project to one dimension and check

- For each cluster, project down to one dimension
  - Use a statistical test to see if the data is Gaussian



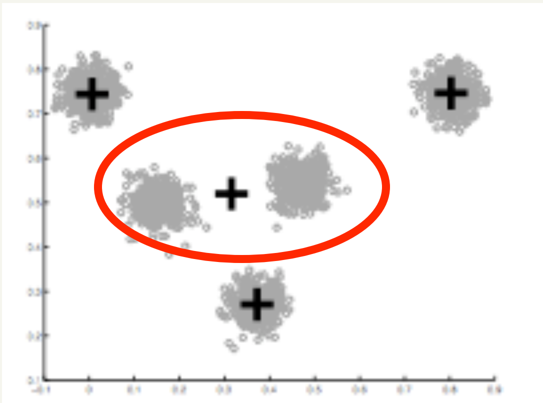
Project to one dimension and check

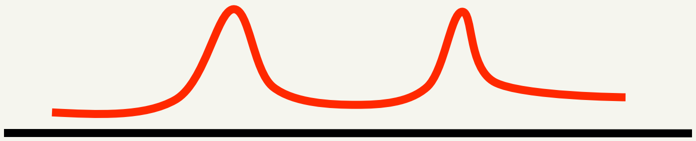
- For each cluster, project down to one dimension
  - Use a statistical test to see if the data is Gaussian



Project to one dimension and check

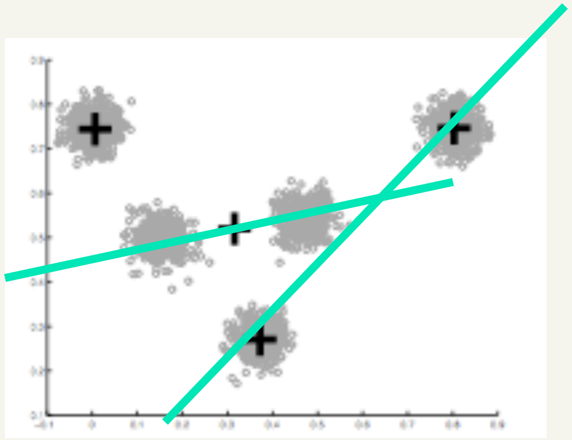
- For each cluster, project down to one dimension
  - Use a statistical test to see if the data is Gaussian





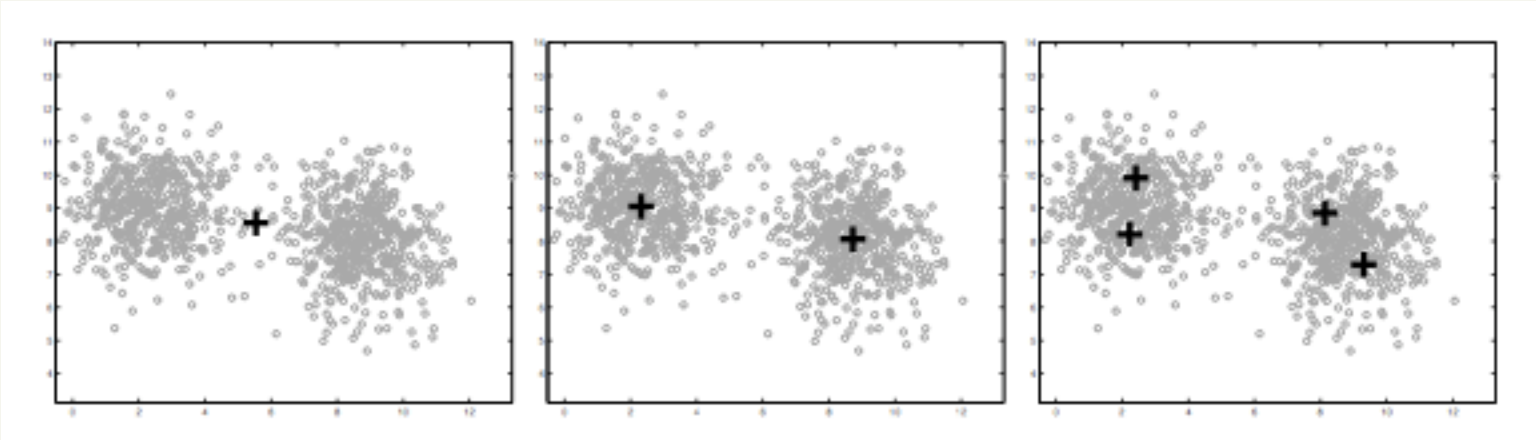
Project to one dimension and check

- For each cluster, project down to one dimension
  - Use a statistical test to see if the data is Gaussian



The dimension of the projection is based on the data

On synthetic data



pass pass fail

Compared to other approaches

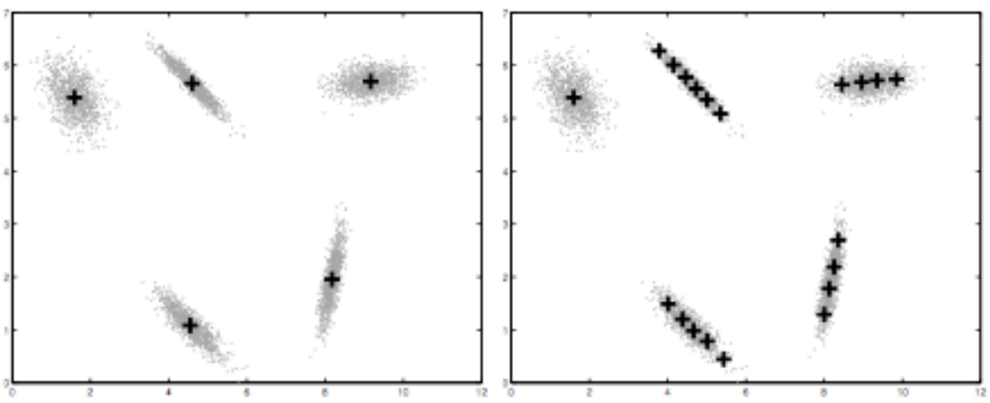


Figure 4: 2-d synthetic dataset with 5 true clusters. On the left, G-means correctly chooses 5 centers and deals well with non-spherical data. On the right, the BIC causes X-means to overfit the data, choosing 20 unevenly distributed clusters.

http://cs.baylor.edu/~hamerly/papers/nips_03.pdf

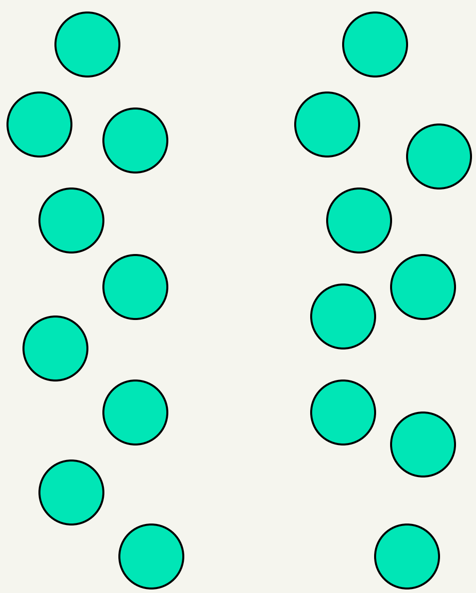
Time Complexity

- Variables: K clusters, n documents, m dimensions, I iterations
- What is the runtime complexity?
  - Computing distance between two docs is O(m) where m is the dimensionality of the vectors.
  - Reassigning clusters: O(Kn) distance computations, or O(Knm)
  - Computing centroids: Each doc gets added once to some centroid: O(nm)
  - Assume these two steps are each done once for I iterations: O$(Iknm)$

In practice, K-means converges quickly and is fairly fast

Problems with K-means

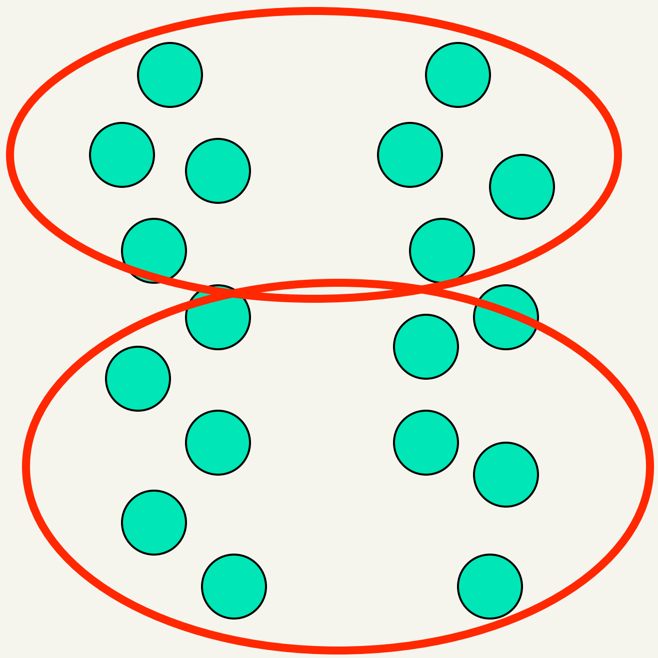
- Determining K is challenging
- Spherical assumption about the data (distance to cluster center)
- Hard clustering isn't always right



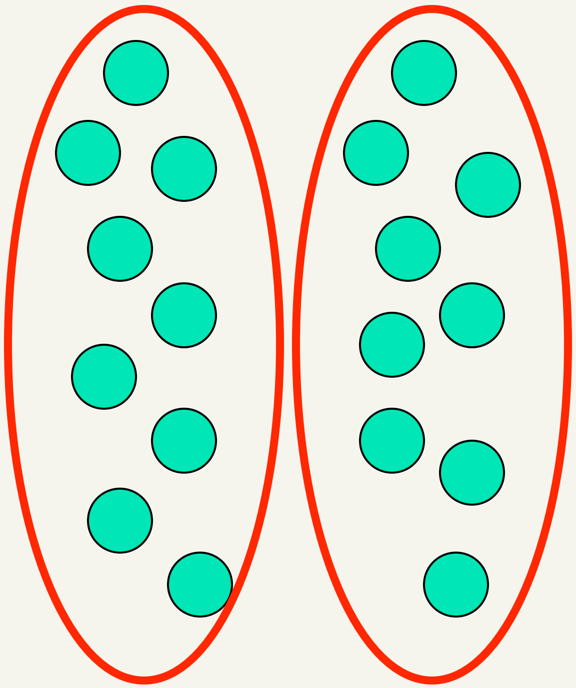
EM clustering: mixtures of Gaussians

Assume data came from a mixture of Gaussians (elliptical data), assign data to cluster with a certain *probability*

k-means



EM



EM is a general framework

- Create an initial model, θ’
  - Arbitrarily, randomly, or with a small set of training examples
- Use the model θ’ to obtain another model θ such that

$$\sum_i \log P_{\theta}(y_i) > \sum_i \log P_{\theta'}(y_i)$$ i.e. better models data

- Let θ’ = θ and repeat the above step until reaching a local maximum
  - Guaranteed to find a better model after each iteration

Where else have you seen EM?

E and M steps

- Use the current model to create a better model

Expectation: Given the current model, figure out the expected probabilities of the documents to each cluster

$$p(x|\theta_c)$$

Maximization: Given the probabilistic assignment of all the documents, estimate a new model, θ_c

Each iterations increases the likelihood of the data and guaranteed to converge!

Similar to *K*-Means

- Iterate:
  - Assign/cluster each document to closest center

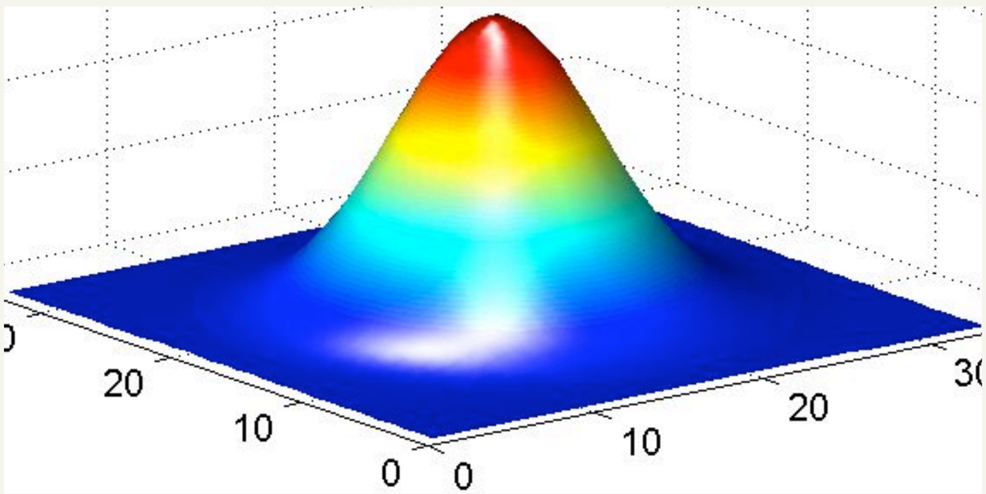
    Expectation: Given the current model, figure out the expected probabilities of the documents to each cluster $p(x|\theta_c)$

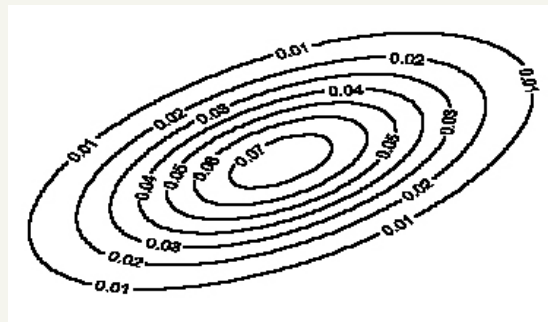
  - Recalculate centers as the mean of the points in a cluster

    Maximization: Given the probabilistic assignment of all the documents, estimate a new model, θ_c

Model: mixture of Gaussians

$$N[x;\mu,\Sigma] = \frac{1}{(2\pi)^{d/2}\sqrt{\det(\Sigma)}}\exp[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)]$$





Covariance determines the shape of these contours

- Fit these Gaussian densities to the data, one per cluster

EM example

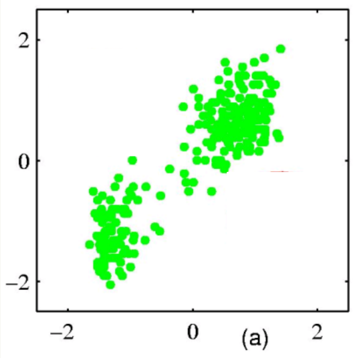


Figure from Chris Bishop

EM example

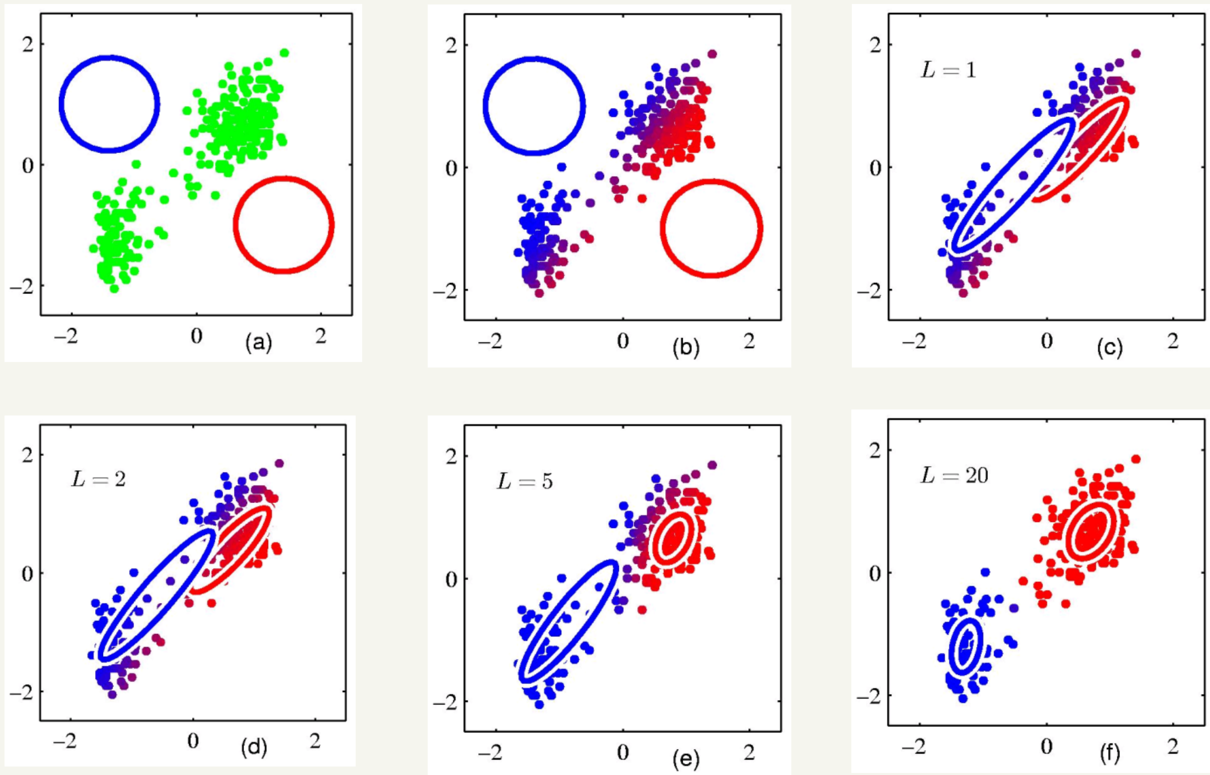
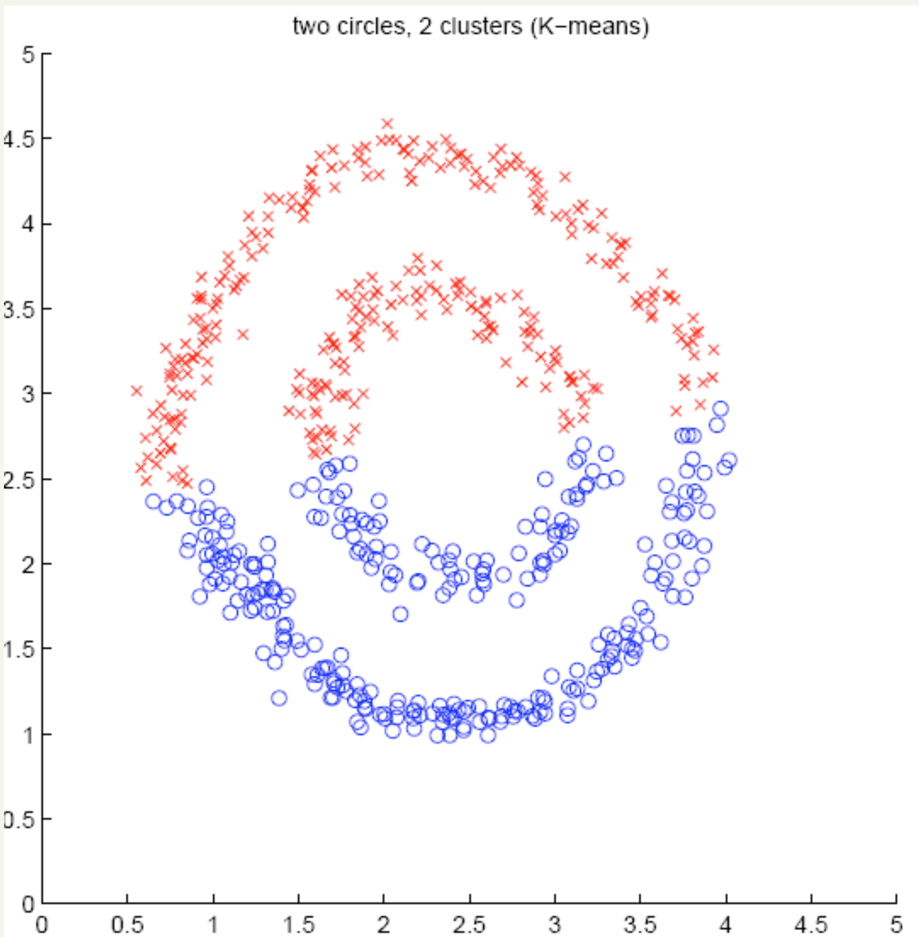


Figure from Chris Bishop

Other algorithms

- K-means and EM-clustering are by far the most popular, particularly for documents
- However, they can’t handle all clustering tasks
- What types of clustering problems can’t they handle?

Non-gaussian data



What is the problem?

Similar to classification: global decision vs. local decision

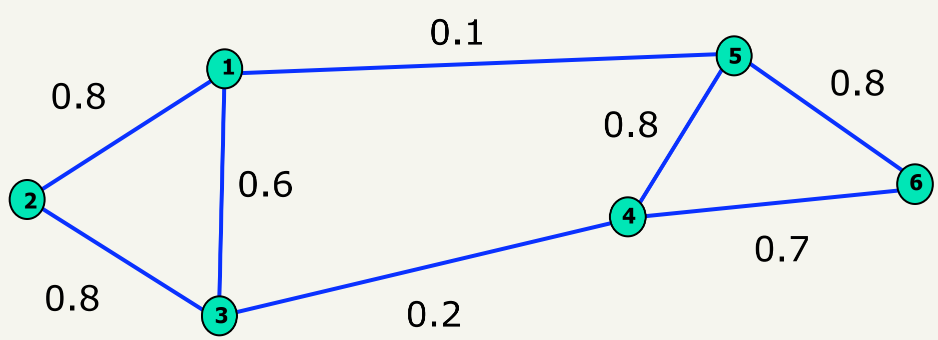
Spectral clustering

Similarity Graph

- Represent dataset as a weighted graph $G(V,E)$
- For documents $\{x_1, x_2, \ldots, x_6\}$

$V=\{x_i\}$ Set of n vertices representing documents/points

$E=\{w_{ij}\}$ Set of weighted edges indicating pair-wise similarity between documents/points

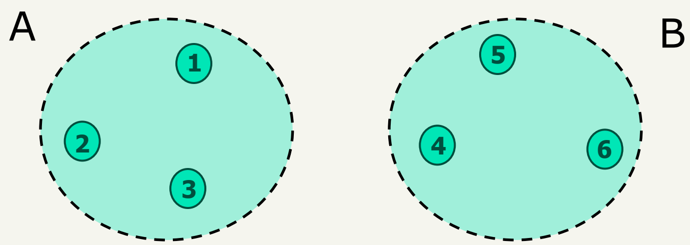


What does clustering represent?

Graph Partitioning

- Clustering can be viewed as partitioning a similarity graph

- *Bi-partitioning* task:
  - Divide vertices into two disjoint groups (A,B)

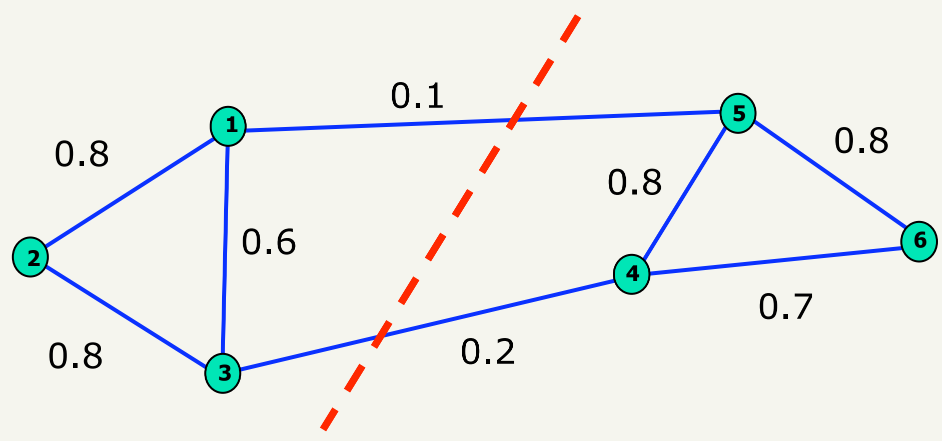


What would define a good partition?

Clustering Objectives

- Traditional definition of a "good" clustering:
  1. Points assigned to same cluster should be highly similar.
  2. Points assigned to different clusters should be highly dissimilar.

- Apply these objectives to our graph representation

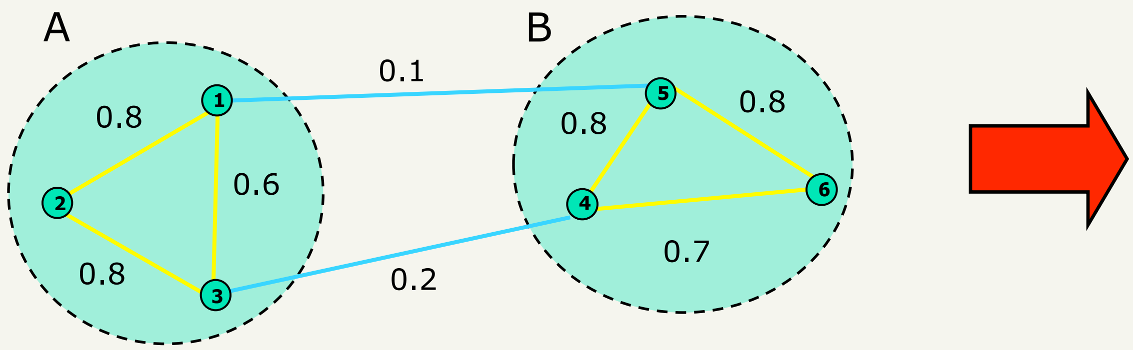


1. Maximise weight of within-group connections
2. Minimise weight of between-group connections

Graph Cuts

- Express partitioning objectives as a function of the "edge cut" of the partition.
- *Cut:* Set of edges with only one vertex in a group.

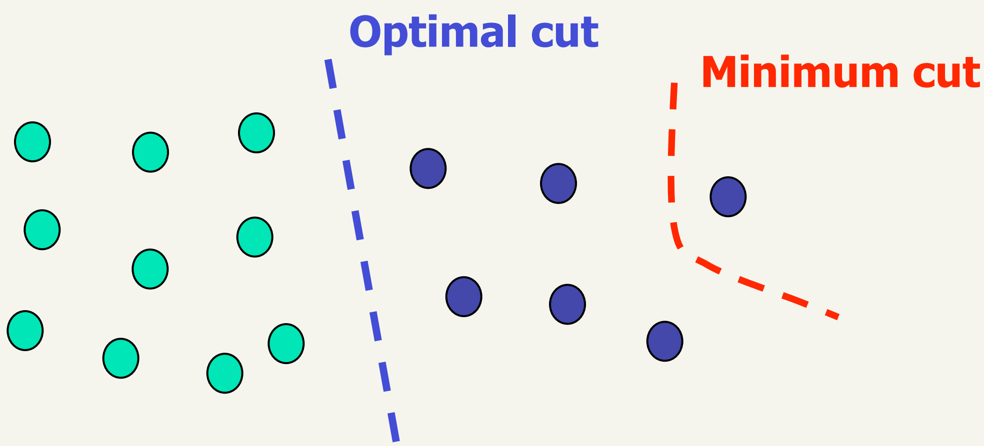
$$cut(A,B) = \sum_{i \in A,\, j \in B} w_{ij}$$



$$cut(A,B) = 0.3$$

Can we use the minimum cut?

Graph Cut Criteria



- **Problem:**
  - Only considers external cluster connections
  - Does not consider internal cluster density

Graph Cut Criteria

- **Criterion: Normalised-cut** (Shi & Malik,'97)
  - Consider the connectivity between groups relative to the density of each group.

$$\min Ncut(A,B) = \frac{cut(A,B)}{vol(A)} + \frac{cut(A,B)}{vol(B)}$$

  - Normalize the association between groups by *volume*
    - $Vol(A)$: The total weight of the edges originating from group A

Why does this work?

Graph Cut Criteria

- **Criterion: Normalised-cut** (Shi & Malik,'97)
  - Consider the connectivity between groups relative to the density of each group.

$$\min Ncut(A,B) = \frac{cut(A,B)}{vol(A)} + \frac{cut(A,B)}{vol(B)}$$

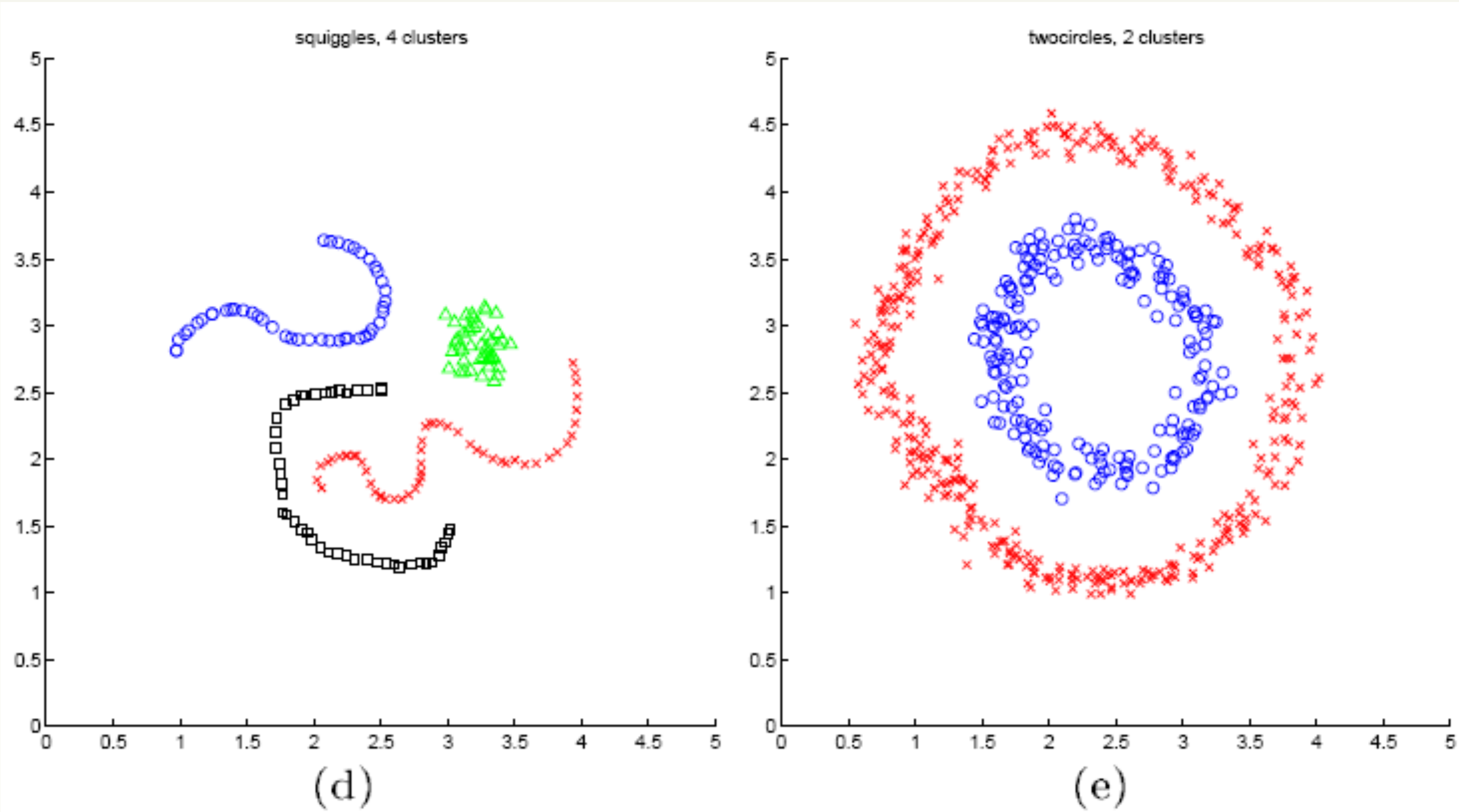
  - Normalize the association between groups by *volume*
    - *Vol(A)*: The total weight of the edges originating from group *A*

Balance between:

Prefers cuts that cut edges on average that are smaller

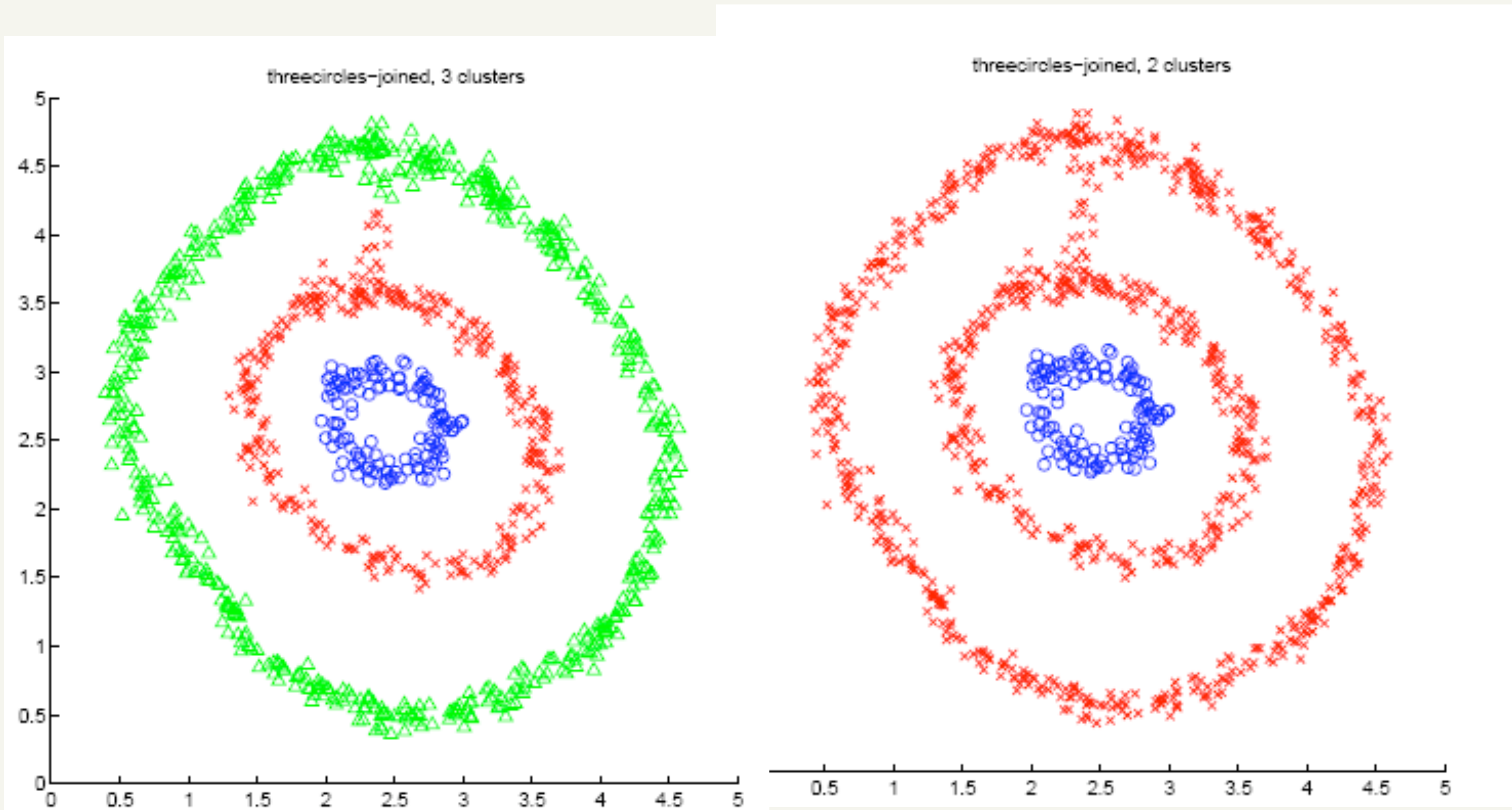
Prefers cuts that cut more edges

Spectral clustering examples



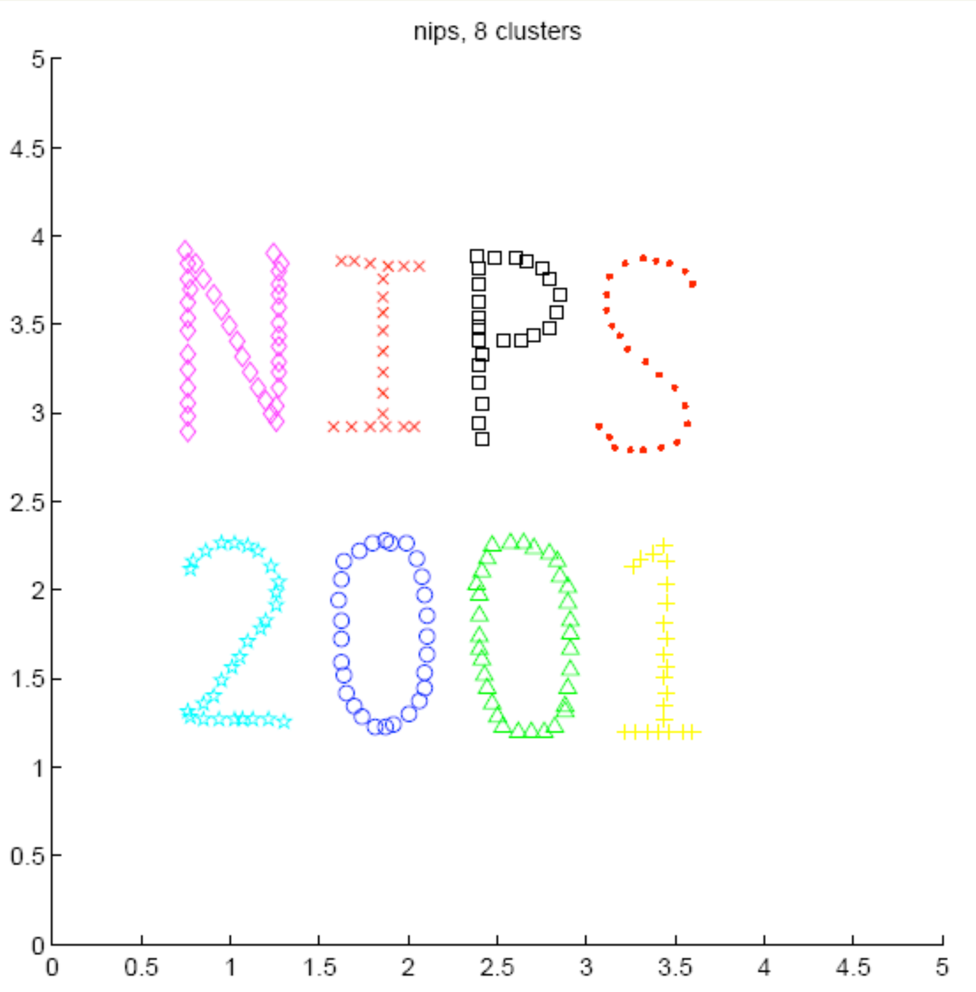
Ng et al On Spectral clustering: analysis and algorithm

Spectral clustering examples



Ng et al On Spectral clustering: analysis and algorithm

Spectral clustering examples



Ng et al On Spectral clustering: analysis and algorithm

Image GUI discussion