clusty.com

Clustering

(1) "Similar" documents should be grouped together

Q: How to assess similarity

(2) Given similarity scores between docs, ~~what algorithm~~ how to generate clustering?

- what sort of clustering?
  - eg. flat clustering (i.e. 10 groups)
    - how many groups?
    - overlapping?
    - every doc in same cluster?
  - eg. hierarchical clustering
    - trees
    - DAGs

judgements.
links to find what want
effort at each level

(1) Some clustering algorithms need
  (a) similarity score
  (b) some need distance

(a): classic solution: cosine similarity in V.S.M.

Scores $\in [0, 1]$

identical docs = 1 ; orthogonal docs = 0

(b) Distance = 1 - similarity (often)

or Euclidean distance in some appropriate model

5-25-2005

Clustering Contd.

Need some similarity measure
- usually just provided. algorithm not concerned w/ it
- usually measures
  - $S(X,X) = 1$
  - $S(X,Y) = S(Y,X)$ symetric
- typically normalized between 0 and 1
- many possibilities
  - Euclidian dist
  - Cos similarity
  - Jaqcard, Dice coefficients

Why Cluster

- alternate representation of data
- unsupervised learning
- can be viewed as dimensionality reduction
- improve efficiency by comparing against clusters not individual docs. not as impt for retrieval today b/c of machine speed
- clustering used in disk layout
- increase effectiveness of retrieval
  - may be better to retrieve clusters to get relevant docs that don't contain as many query terms

"Cluster Hypothesis" - closely associated documents tend to be relevant to same queries

Clustering Algorithms

- many, many algorithms
- virtually no "right" way to do clustering

2 categories (primarily)

- Graph Theoretic
  - define docs as nodes in graph, edges are similarity values
    - complete, undirected graph
    - every node connected to every edge # edges $= \binom{n}{2} = \frac{n(n-1)}{2} \approx n^2$
    - naturally hierarchical
    - structure often too large

- Cluster Representations
  - decide apriori # of clusters or cluster seeds
  - usualy produce partitions
  - usualy run in $O(n)$ or $O(n \log n)$

Graph Theoretic Aproaches - assume graph of objects connected by links w/ similarity greater than some threshold

Single link - if A & B are connected they should be in the same cluster

Complete link - all items in cluster must be connected

Average link - all cluster members most have a greater avg similarity to other cluster members than avg similarity to any other group

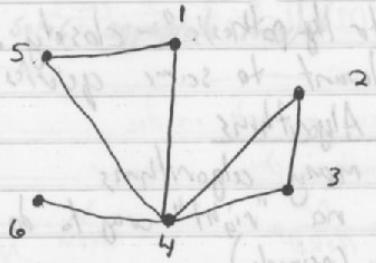
Star Clustering - find greatest # links + cluster then find next greatest # links + cluster

Example

Pick threshold: 0.65

	1	2	3	4	5
1					
2	.6				
3	.6	.8			
4	.9	.7	.7		
5	.9	.6	.6	.9	
6	.5	.5	.5	.9	.5

Threshold Similarity Graph



Clusters:

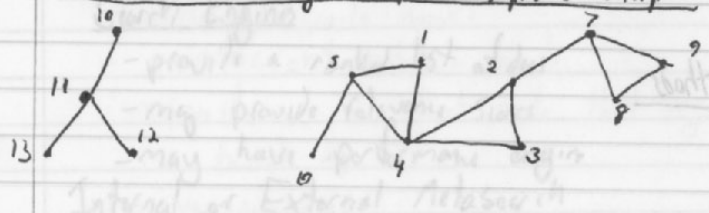
① Single Link: - one cluster (everyone is connected) {1, 2, 3, 4, 5, 6}

② Complete Link - {{1, 4, 5}, {2, 3, 4}, {4, 6}}

③ Star Clustering {1, 2, 3, 4, 5, 6}
└ star center

can choose overlapping or non-overlapping

Threshold Similarity Graph (separate clusters)



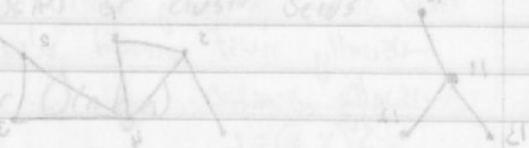
① Single Link: 2 clusters
{(1, 2, ... 9), (10, 11, 12, 13)}

② Complete Link: {(1, 4, 5), (2, 3, 4), (2, 8, 9), (2, 7), (5, 6), (10, 1), (11, 12), (11, 13)}
8 clusters

③ Star Clusters: {(1, 2, 3, 4, 5), (7, 8, 9, 2), (11, 12, 13, 10), (6, 5)}

5-26-2005 Clustering Cont:

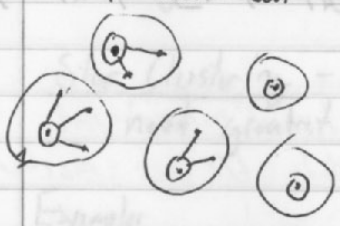
Fast Partition Methods



Basic Idea
- select representatives
- cluster reps w/ non-closest rep

Q
- How to pick reps?
- How many reps?
- How to cluster w/ reps?

Algorithm 1
~~K-means but w/ threshold of min similarity~~



Running time: $O((\#\text{reps}) * (\#\text{obj}))$
- worst case is # reps = # objects = n^2

- variant: recalc centroid each time a cluster is modified
- uses threshold to compute reps on fly

K-means
- how to pick K-reps

How to find "good" representatives?
- one good answer: furthest first traversal
  - pick initial point
  - for each subsequent point, pick one which is furthest away
    - point who's minimum distance to any point is largest

eg.