PURDUE UNIVERSITY

CS47300: Web Information Search and Management

Text Clustering

Prof. Chris Clifton

17 October 2019

Borrows slides from Chris Manning, Ray Mooney and Soumen Chakrabarti

PURDUE UNIVERSITY

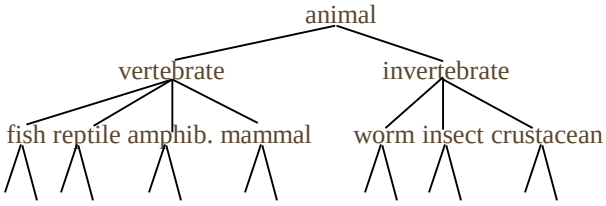
Soft Clustering

- Clustering typically assumes that each instance is given a "hard" assignment to exactly one cluster.
- Does not allow uncertainty in class membership or for an instance to belong to more than one cluster.
- *Soft clustering* gives probabilities that an instance belongs to each of a set of clusters.
- Each instance is assigned a probability distribution across a set of discovered categories (probabilities of all categories must sum to 1).

PURDUE UNIVERSITY

Hierarchical Clustering

- Build a tree-based hierarchical taxonomy (*dendrogram*) from a set of unlabeled examples.



- One option to produce a hierarchical clustering is recursive application of a partitional clustering algorithm to produce a hierarchical clustering.

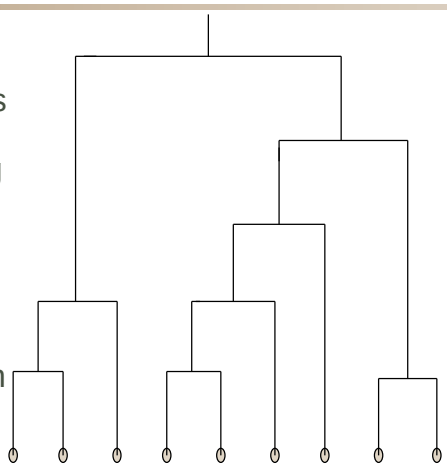
PURDUE UNIVERSITY

Hierarchical Agglomerative Clustering (HAC)

- Assumes a similarity function for determining the similarity of two instances.
- Starts with all instances in a separate cluster and then repeatedly joins the two clusters that are most similar until there is only one cluster.
- The history of merging forms a binary tree or hierarchy.

A Dendogram: Hierarchical Clustering

- Dendrogram: Decomposes data objects into a several levels of nested partitioning (tree of clusters).
- Clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.

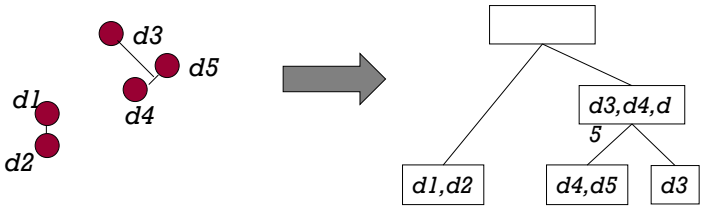


Hierarchical Clustering algorithms

- **Agglomerative (bottom-up):**
  - Start with each document being a single cluster.
  - Eventually all documents belong to the same cluster.
- **Divisive (top-down):**
  - Start with all documents belong to the same cluster.
  - Eventually each node forms a cluster on its own.
- Does not require the number of clusters ***k*** in advance
- Needs a termination/readout condition
  - The final mode in both Agglomerative and Divisive is of no use.

Dendrogram: Document Example

- As clusters *agglomerate*, docs likely to fall into a hierarchy of "topics" or concepts.



"Closest pair" of clusters

- Many variants to defining closest pair of clusters
- "Center of gravity"
  - Clusters whose centroids (centers of gravity) are the most cosine-similar
- Average-link
  - Average cosine between pairs of elements
- Single-link
  - Similarity of the most cosine-similar (single-link)
- Complete-link
  - Similarity of the "furthest" points, the least cosine-similar

PURDUE UNIVERSITY

Hierarchical Clustering

- Key problem: as you build clusters, how do you represent the location of each cluster, to tell which pair of clusters is closest?
- Euclidean case: each cluster has a *centroid* = average of its points.
  - Measure intercluster distances by distances of centroids.

PURDUE UNIVERSITY

Single Link Agglomerative Clustering

- Use maximum similarity of pairs:

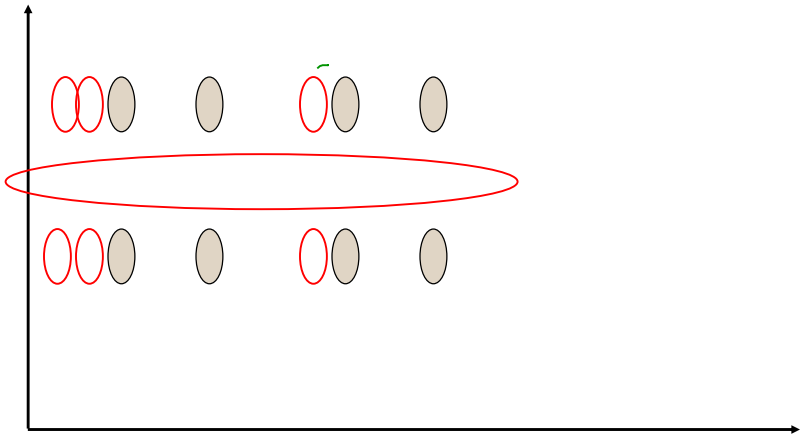
$$sim(c_i, c_j) = \max_{x \in c_i, y \in c_j} sim(x, y)$$

- Can result in "straggly" (long and thin) clusters due to chaining effect.
  - Appropriate in some domains, such as clustering islands: "Hawaii clusters"
- After merging c_i and c_j, the similarity of the resulting cluster to another cluster, c_k, is:

$$sim((c_i \cup c_j), c_k) = \max(sim(c_i, c_k), sim(c_j, c_k))$$

PURDUE UNIVERSITY

Single Link Example



PURDUE UNIVERSITY

Complete Link Agglomerative Clustering

- Use minimum similarity of pairs:

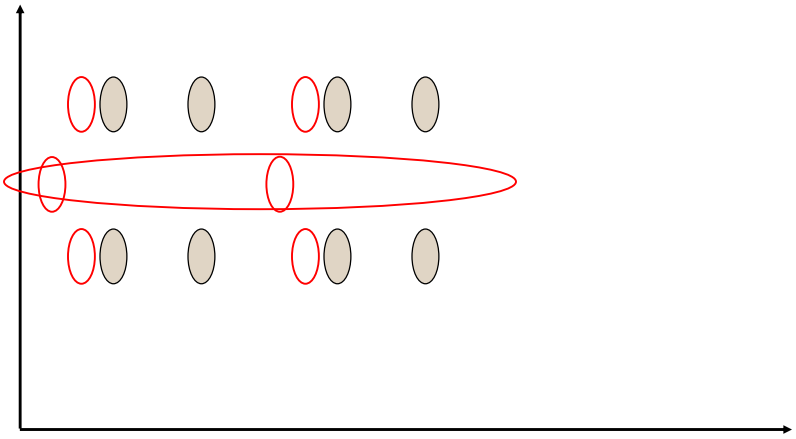
$$sim(c_i, c_j) = \min_{x \in c_i, y \in c_j} sim(x, y)$$

- Makes "tighter," spherical clusters that are typically preferable.
- After merging c_i and c_j, the similarity of the resulting cluster to another cluster, c_k, is:

$$sim((c_i \cup c_j), c_k) = \min(sim(c_i, c_k), sim(c_j, c_k))$$

PURDUE UNIVERSITY

Complete Link Example



PURDUE UNIVERSITY

Evaluation of clustering

- Perhaps the most substantive issue in data mining in general:
  - how do you measure goodness?
- Most measures focus on computational efficiency
  - Time and space
- For application of clustering to search:
  - Measure retrieval effectiveness

PURDUE UNIVERSITY

Approaches to evaluating

- Anecdotal
- User inspection
- Ground “truth” comparison
  - Cluster retrieval
- Purely quantitative measures
  - Probability of generating clusters found
  - Average distance between cluster members
- Microeconomic / utility

PURDUE UNIVERSITY

Anecdotal evaluation

- Probably the commonest (and surely the easiest)
  - “I wrote this clustering algorithm and look what it found!”
- No benchmarks, no comparison possible
- Any clustering algorithm will pick up the easy stuff like partition by languages
- Generally, unclear scientific value.

PURDUE UNIVERSITY

User inspection

- Induce a set of clusters or a navigation tree
- Have subject matter experts evaluate the results and score them
  - some degree of subjectivity
- Often combined with search results clustering
- Not clear how reproducible across tests.
- Expensive / time-consuming

PURDUE UNIVERSITY

Ground “truth” comparison

- Take a union of docs from a taxonomy & cluster
  - Yahoo!, ODP, newspaper sections …
- Compare clustering results to baseline
  - e.g., 80% of the clusters found map “cleanly” to taxonomy nodes
  - How would we measure this?
- But is it the “right” answer?
  - There can be several equally right answers “Subjective”
- For the docs given, the static prior taxonomy may be incomplete/wrong in places
  - the clustering algorithm may have gotten right things not in the static taxonomy

PURDUE UNIVERSITY

Evaluation example: Cluster retrieval

- Ad-hoc retrieval
- Cluster docs in returned set
- Identify best cluster & only retrieve docs from it
- How do various clustering methods affect the quality of what's retrieved?
- Concrete measure of quality:
  - Precision as measured by user judgements for these queries
- Done with TREC queries

PURDUE UNIVERSITY

Evaluation

- Compare two IR algorithms
  - 1. send query, present ranked results
  - 2. send query, cluster results, present clusters
- Experiment was simulated (no users)
  - Results were clustered into 5 clusters
  - Clusters were ranked according to percentage relevant documents
  - Documents within clusters were ranked according to similarity to query

PURDUE UNIVERSITY

Sim-Ranked vs. Cluster-Ranked

	Precision at Cutoffs		
CutOff	Sim-Ranked	Cluster-Ranked	% Increase
5	.342	.428	.252
10	.314	.401	.277
20	.276	.363	.312

Table 4: Precision at small document cutoff levels for the one-step algorithm.

PURDUE UNIVERSITY

"The Curse of Dimensionality"

- Why document clustering is difficult
  - While clustering looks intuitive in 2 dimensions, many of our applications involve 10,000 or more dimensions…
  - High-dimensional spaces look different: the probability of random points being close drops quickly as the dimensionality grows.
  - One way to look at it: in large-dimension spaces, random vectors are almost all almost perpendicular. Why?
- Solution: Dimensionality reduction … important for text

PURDUE UNIVERSITY

Related Tasks

- TDT
  - Topic Detection: “Dynamic” Clustering
  - Topic Tracking: on-line categorization
  - Story Segmentation
  - First Story Detection
  - New Information Detection
  - Story Link Detection
- TIDES
  - All of the above in multilingual and multimedia
- Word cloud
- And others…