A Comparison of Document Clustering Techniques

M. Steinbach, G. Karypis, V. Kumar

Present by Leo Chen
Road Map

• Background & Motivation (2)

• Basic (6)
  – Vector Space Model
  – Cluster Quality Evaluation

• Agglomerative Clustering Algorithm (4)

• K-means & Bisecting K-means (6)

• Comparisons & Explanations (5)

• Conclusions (3)
Background & Motivation

• Wide usage of document clustering
  – Improving the precision in information retrieval system [Rij79, Kow97] (how?)
  – Finding the nearest neighbors of a document [BL85]
  – Browsing a collection of documents [CKPT92]
  – Organizing the results returned by a search engine [ZEMK97]
  – Generating hierarchical clusters of document [KS97]
  – Producing an effective document classifier [AGY99]

• Agglomerative hierarchical clustering & K-means
  – Agglomerative hierarchical clustering is better? [DJ88], [CKPT92], [LA99]
  – A simple and efficient variant of K-means is better? Follow me!
Vector Space Model

- Document vector: \( D_{tf} = (t_{f1}, t_{f2}, \ldots, t_{fn}) \)
- \( t_{fi} \): frequency of the \( i^{th} \) term in the document
- Weight each term based on its inverse document frequency (how? Why?)
- Normalize each vector to unit length: \( \|d\| = 1 \)
- \( \text{cosine}(d_1, d_2) = (d_1 \cdot d_2) / (\|d_1\| \cdot \|d_2\|) = d_1 \cdot d_2 \)
- Centroid: \( c = \Sigma d / |S| \)
- Average pairwise similarity:
  \( \Sigma \text{cosine}(d', d) / |S|^2 = \Sigma d' \cdot d / |S|^2 \)
  \( = (\Sigma d' / |S|) \cdot (\Sigma d / |S|) \)
  \( = c \cdot c = \| c \|^2 \)
Cluster Quality Evaluation (1)

- **Entropy** \([\text{Sha48}]\) (the lower, the better)
  - Class distribution:
    - \(p_{ij}\), the “probability” that a member of cluster \(j\) belongs to class \(i\).
  - Entropy of cluster \(j\):
    \(E_j = - \sum p_{ij} \log (p_{ij})\)
  - Total Entropy:
    \(E_{cs} = \sum (n_j \times E_j / n)\)
Cluster Quality Evaluation (2)

- **F-measure** [LA99] (the higher, the better)
  
  - For cluster \( j \) and class \( i \),
    
    \[
    \text{Recall} (i, j) = \frac{n_{ij}}{n_i}, \quad \text{Precision} (i, j) = \frac{n_{ij}}{n_j}
    \]
  
  - \( F (i, j) = \frac{2 \times \text{Recall} (i, j) \times \text{Precision} (i, j)}{(\text{Precision} (i, j) + \text{Recall} (i, j))} \)
  
  - Entire F-measure: \( F = \sum \max \{ F (i, j) \} \times \frac{n_i}{n} \)

- **Overall similarity:** \( \| c \|^2 \) (the higher, the better)
Agglomerative Clustering Algorithm (1)

- **Simple Agglomerative Clustering Algorithm**
  1. Compute similarity between all pairs of clusters
  2. Merge the most similar (closest) two clusters
  3. Update the similarity matrix
  4. Repeat 2 & 3 until only a single cluster remains
Agglomerative Clustering Algorithm (2)

• Comparison of Agglomerative Clustering Algorithm
  – IST: Intra-Cluster Similarity
    look at the similarity of all the documents in a cluster to their cluster centroid
  – CST: Centroid Similarity Technique
    look at the cosine similarity between the centroids of the two clusters
  – UPGMA: [DJ88, KR90]
    look at cluster similarity as following:
    \[
    \text{similarity (cluster}_1, \text{ cluster}_2) = \sum \text{cosine (d}_1, \text{ d}_2) / (\text{size (cluster}_1) * \text{size (cluster}_2))
    \]
Agglomerative Clustering Algorithm (3)

- Experiment Results Shows, the winner of the three agglomerative hierarchical techniques is: **UPGMA!**
K-means & Bisecting K-means (1)

- Basic K-means Algorithm
  1. Select K points as the initial centroids
  2. Assign all points to the closest centroid
  3. Recompute the centroid of each cluster
  4. Repeat steps 2 & 3 until the centroids don’t change
K-means & Bisecting K-means (2)

- Basic Bisecting K-means Algorithm
  1. Pick a cluster to split (split the largest
  2. Find 2 sub-clusters using the basic K-means algorithm
  3. Repeat step 2, the bisecting step, for ITER times and take the split that produces the clustering with the highest overall similarity
  4. Repeat steps 1, 2 and 3 until the desired number of clusters is reached
  5. Divisive hierarchical clustering algorithm.

Complexity?
Comparison & Explanations (1)

- Bisecting K-means, with or without refinement is better than regular K-means and UPGMA, with or without refinement, in most cases.
- Refinement significantly improve the performance of UPGMA for both the overall similarity and the entropy measures.
- Regular K-means, is generally better than UPGMA.
Comparison & Explanations (2)

• Why agglomerative hierarchical clustering performs poorly?
  – Documents share “core” vocabularies.
  – Two documents can often be nearest neighbors without belonging to the same class, so agglomerative algorithms make mistakes.
  – “Global properties” help overcome local minima.
  – K-means better suited to document clustering.
Comparison & Explanations (3)

- Why bisecting K-means works better than regular K-means?
  - Bisecting K-means tends to produce clusters of relatively uniform size.
  - Regular K-means tends to produce clusters of widely different sizes.
  - Bisecting K-means beats Regular K-means in Entropy measurement.
Conclusions

- K-means, improvement
  - Many runs
  - Incremental updating
  - Hybrid approach
- My doubts?
  - Measurement?
  - Complexity?
  - Bona fide K-means?
  - Applicable scope?