Document clustering with committees

報告者：黃立德

References:

Patrick Pantel and Dekang Lin, “Document Clustering with Committees”, SIGIR’02

Dan Siroker and Steve Miller, “Topical Clustering, Summarization, and Visualization”

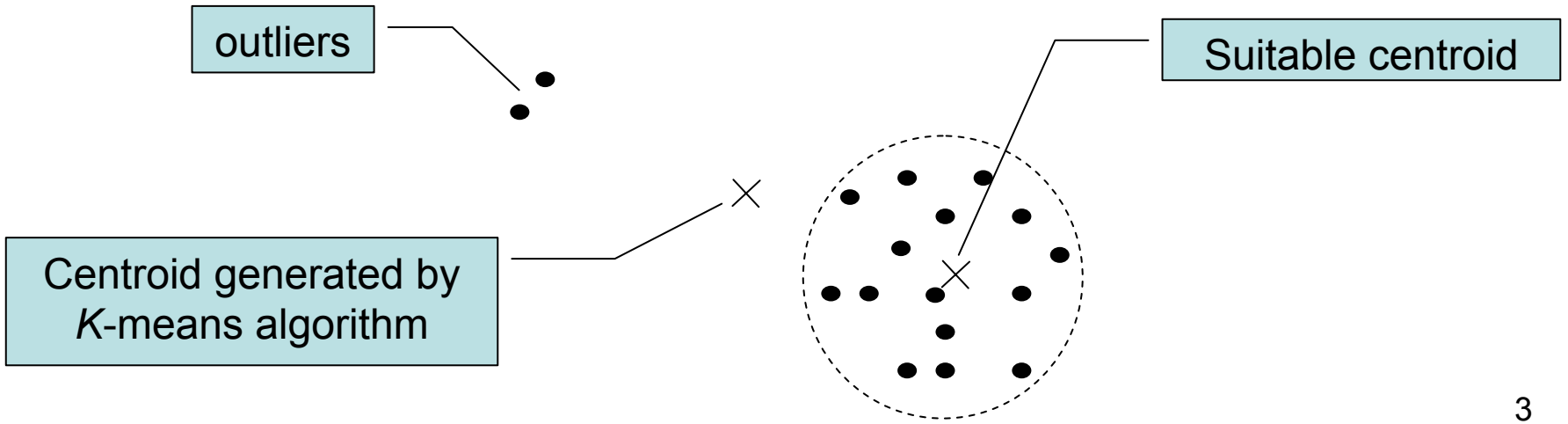
Berlin Chen, slides of “Machine Learning and Data Mining”, 2004

Outline

- Drawbacks of *K*-means
- Concepts of CBC algorithm
- Observations of CBC's experimental results
- HAC-based improvement
- Topic detection
- Conclusions & Demos

Drawbacks of *K*-means

- The qualities of clusters may be different because of initial centroids and the value of *K*
- The centroid of a cluster may be unduly influenced by documents that only marginally belong to the cluster (outliers)

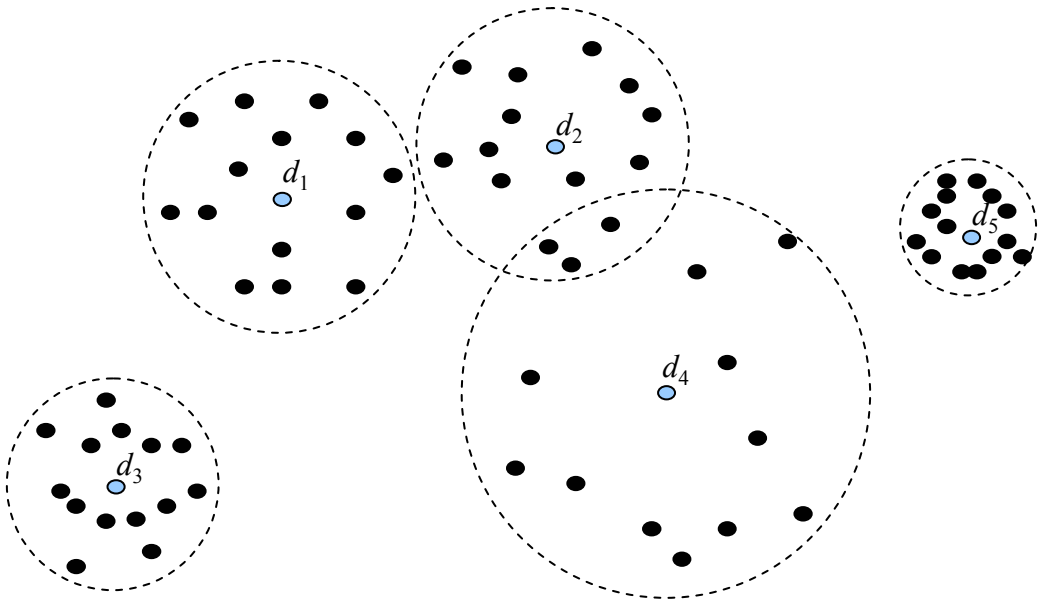


Concepts of CBC algorithm

- CBC (Clustering By Committee) algorithm
  - It initially discovers a set of tight clusters (high intra-group similarity), called committees
  - The committees are well scattered in the feature space (low inter-group similarity)
  - The number of clusters is not fixed
  - The centroids do not change while adding a document to a cluster

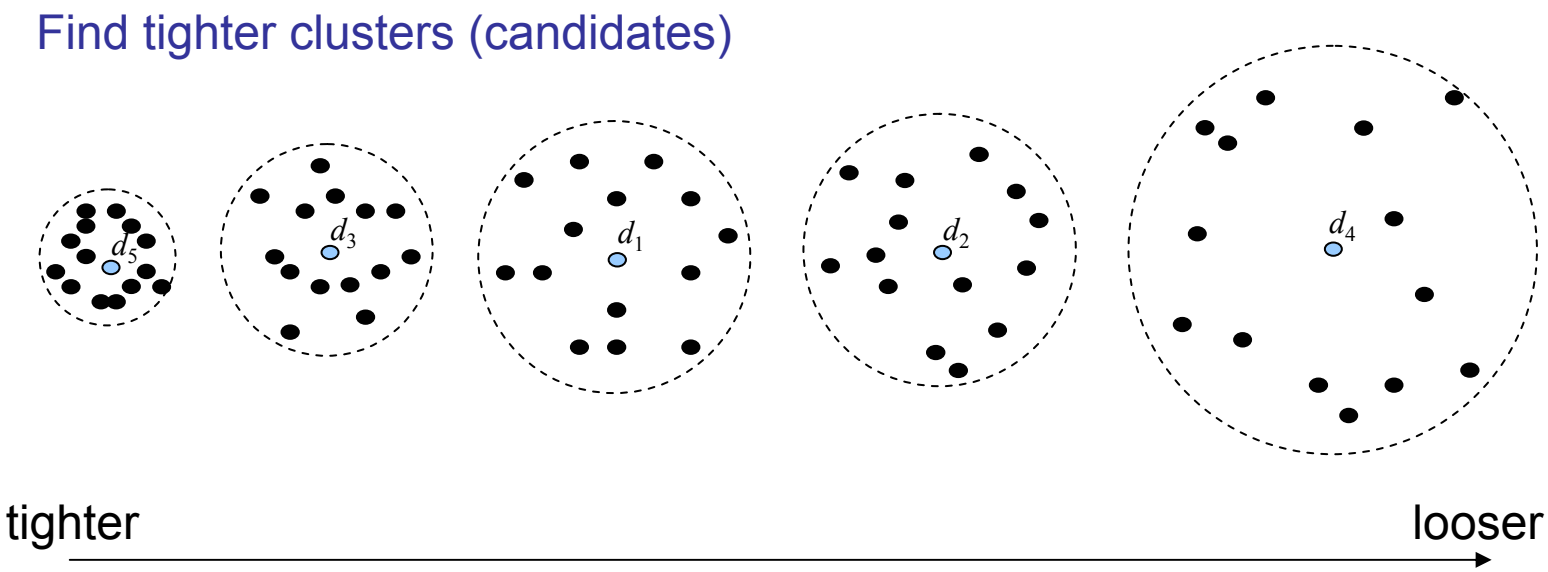
Visualization of CBC algorithm

- Step 1. Find top-similar documents



Visualization of CBC algorithm (cont.)

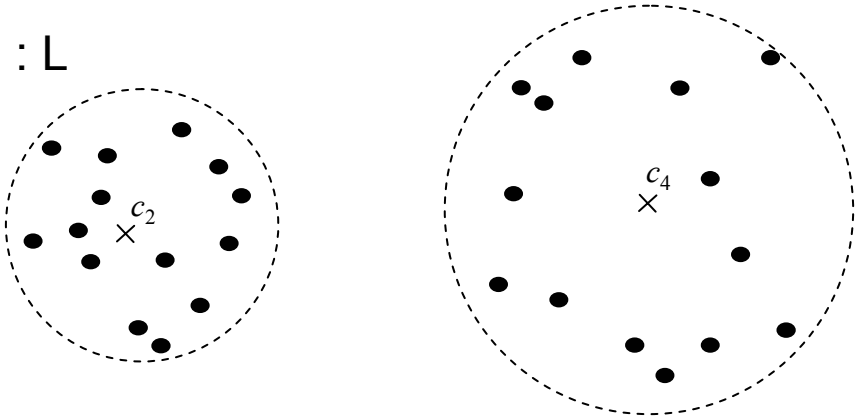
- Step 2. Find committees



Visualization of CBC algorithm (cont.)

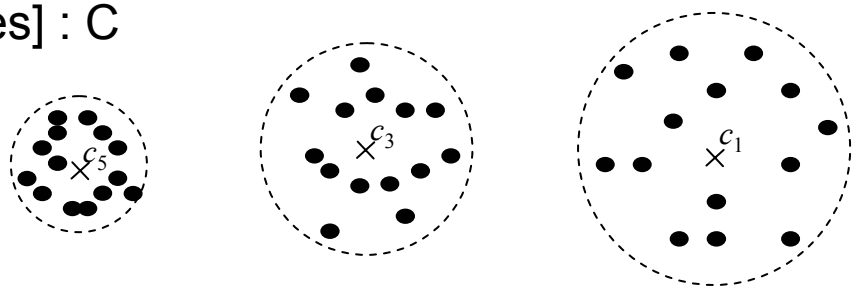
Find committees

[candidates] : L



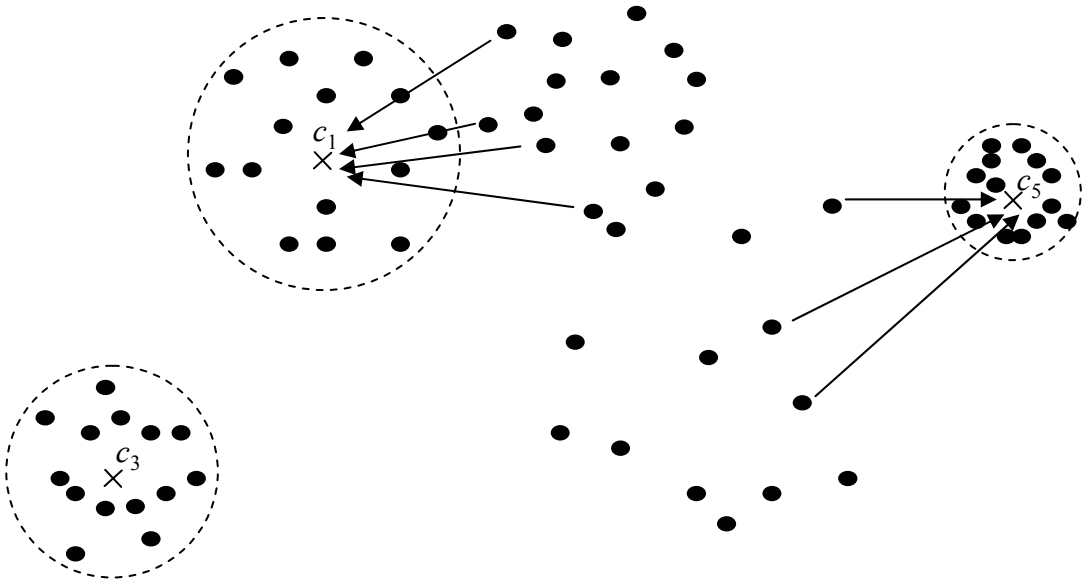
If one's similarity to the centroid of each committee previous added to C is below a threshold, add it to C

[committees] : C



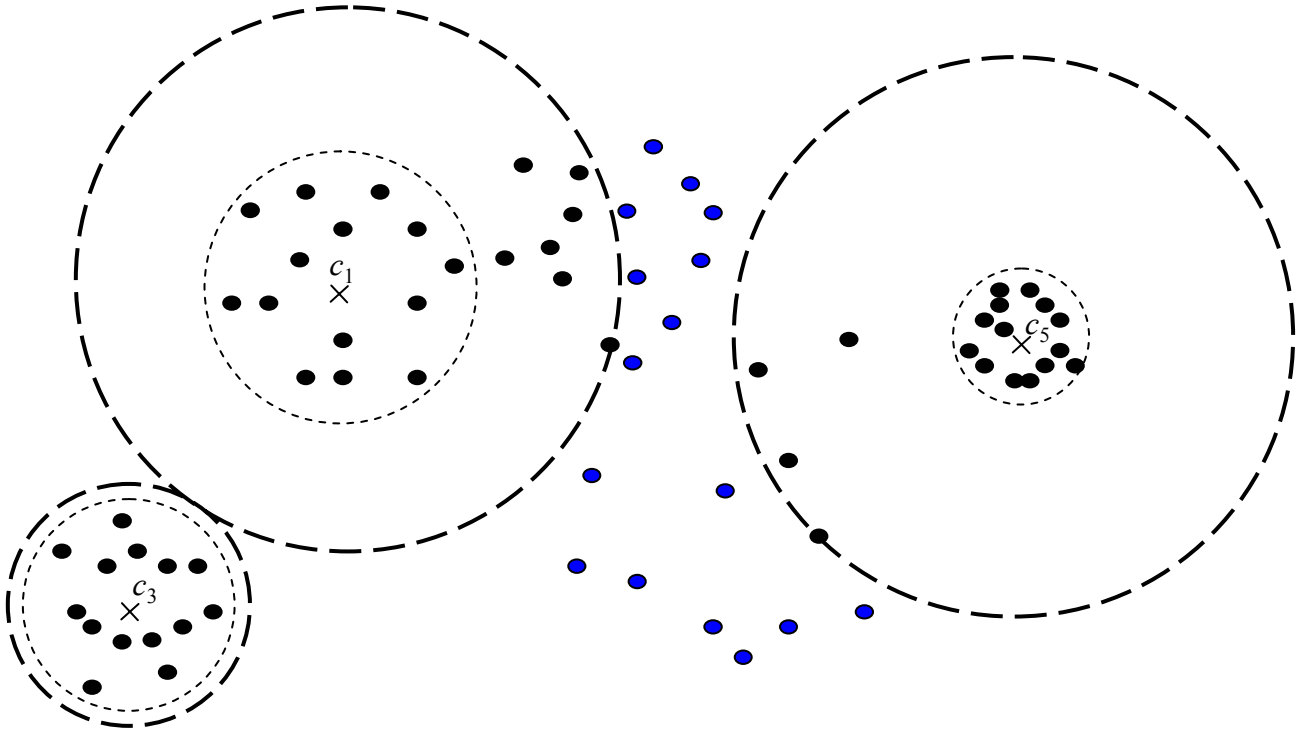
Visualization of CBC algorithm (cont.)

Identify residue documents (1)



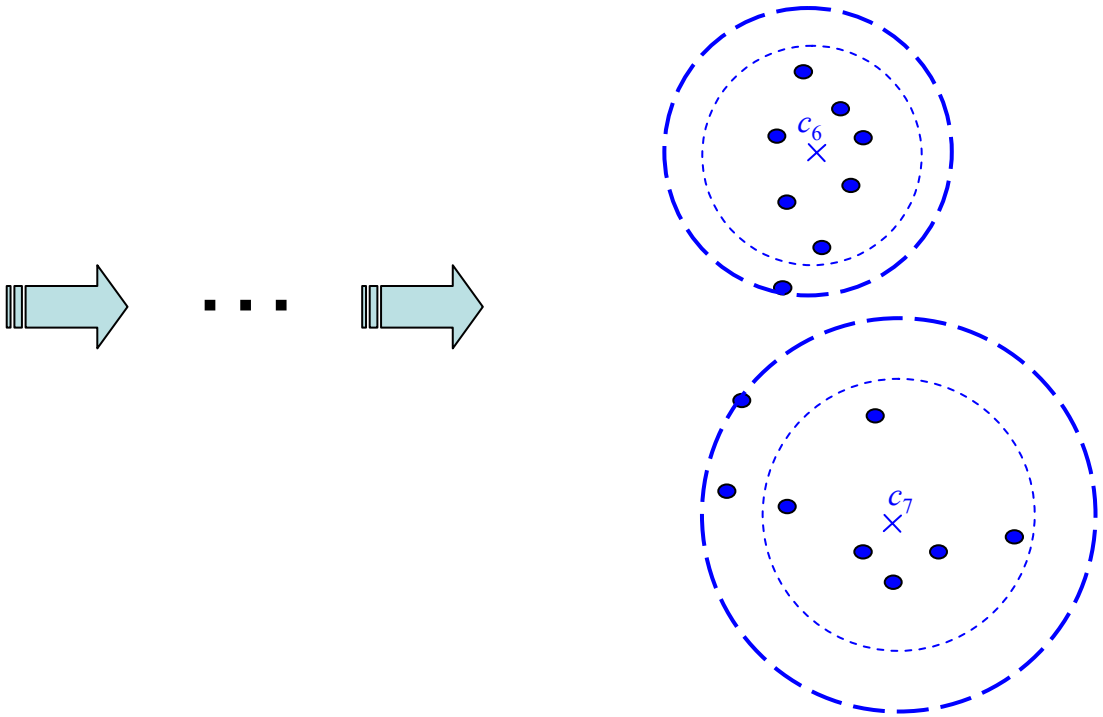
Visualization of CBC algorithm (cont.)

Identify residue documents (2)



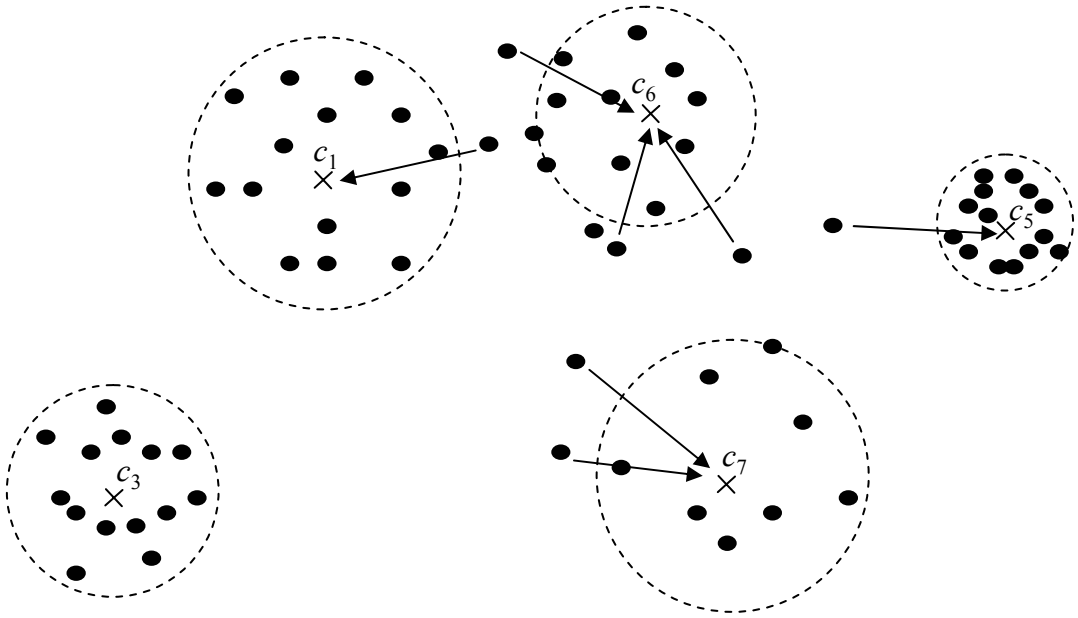
Visualization of CBC algorithm (cont.)

Find committees recursively (3)



Visualization of CBC algorithm (cont.)

- Step 3. Assign documents to clusters



Representation

- Represent a document as a mutual information vector $MI(e) = (mi_{e1}, mi_{e2}, \ldots, mi_{em})$,where mi_{ef} is the mutual information between document e and feature f , which is defined as

$$mi_{ef} = \log \frac{\frac{c_{ef}}{N}}{\frac{\sum_i c_{if}}{N} \times \frac{\sum_j c_{ej}}{N}} \quad , where\ N = \sum_i \sum_j C_{ij}$$

Representation (cont.)

- Compute the similarity between two documents e_i and e_j using the cosine coefficient

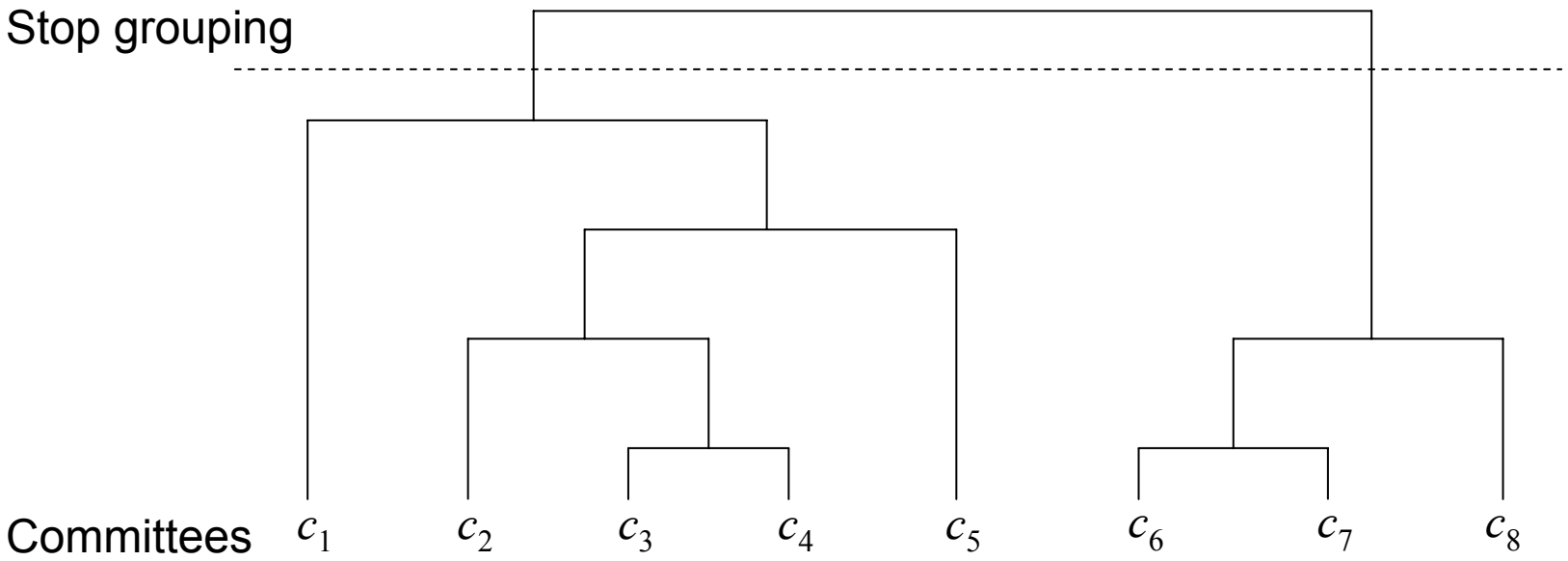
$$sim(e_i, e_j) = \frac{\sum_f mi_{e_i f} \times mi_{e_j f}}{\sqrt{\sum_f mi_{e_i f}^2 \times \sum_f \times mi_{e_j f}^2}}$$

Observations of CBC's experimental results

- It determines the number of committees automatically and produce high qualities of clusters
- It spends most of time to find committees
- The number of committees is usually large
- It is suitable for clustering events rather than topics as a result of committees

HAC-based improvement

- Bottom-up clustering (hierarchical clustering)
- Start with committees (their means) and group the most similar ones

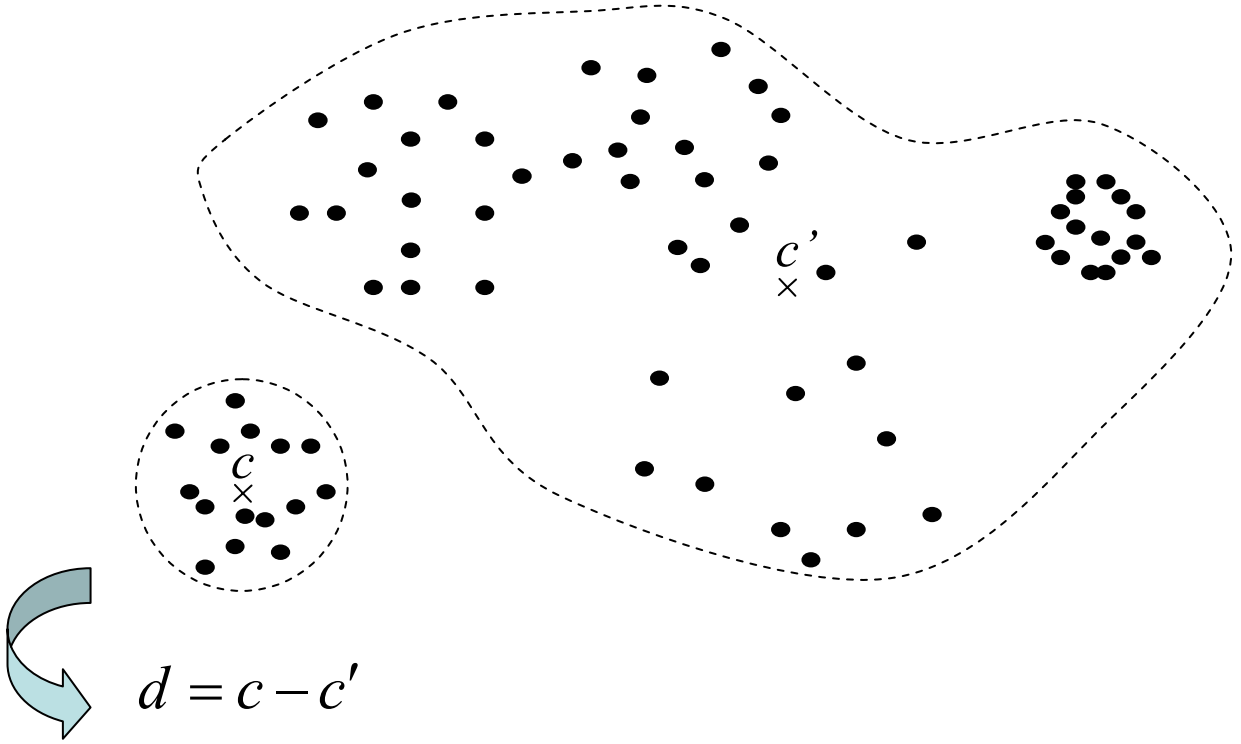


Results of HAC-based improvement

- The executing time is long while there are a lot of committees
- By observing final clusters, we would find that some clusters are too large
- It is a critical issue to decide the final number of clusters

Topic detection

- Maximum vector difference
  - A simple and efficient method



$$d = c - c'$$

$$topic = feature\left(\max\left(d_i\right)\right) \quad , where\ d_i\ is\ an\ entry\ of\ d$$

Conclusions & Demos

- It takes much time to executing CBC algorithm, but we can get good clusters
- Clusters produced by CBC exhibit events and topics were shown after HAC-based improvement
- “Maximum vector difference” for labeling clusters generates qualitatively feasible results in reasonable time