CMU SCS

NOT in the midterm exam

15-826: Multimedia Databases and Data Mining

Lecture #16: Text - part III:
Vector space model and clustering
C. Faloutsos

NOT in the midterm exam

Must-Read Material

- MM Textbook, Chapter 6

CMU SCS

Outline

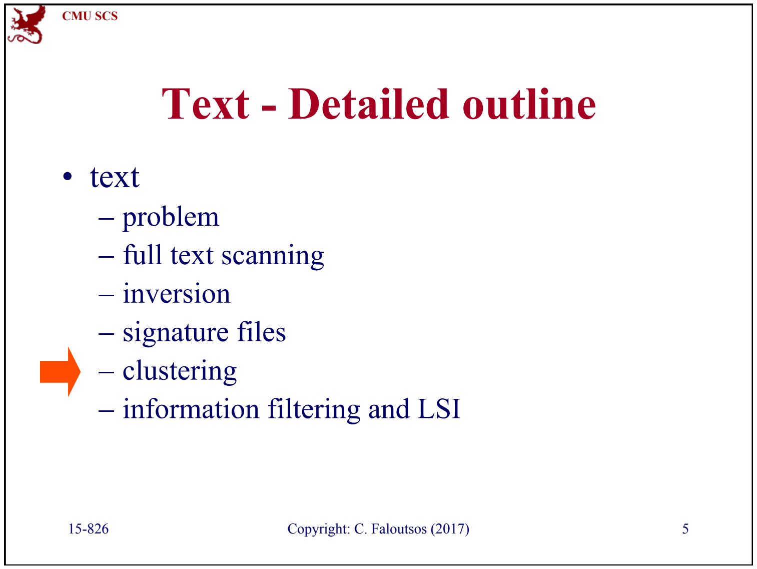
Goal: ‘Find similar / interesting things’

- Intro to DB
- ➡ Indexing - similarity search
- Data Mining

CMU SCS

Indexing - Detailed outline

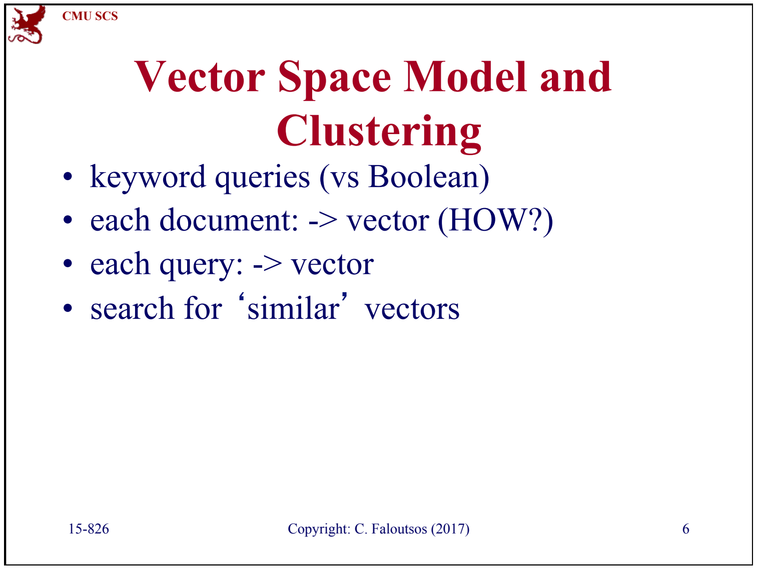
- primary key indexing
- secondary key / multi-key indexing
- spatial access methods
- fractals
- ➡ text
- multimedia
- ...



CMU SCS

Text - Detailed outline

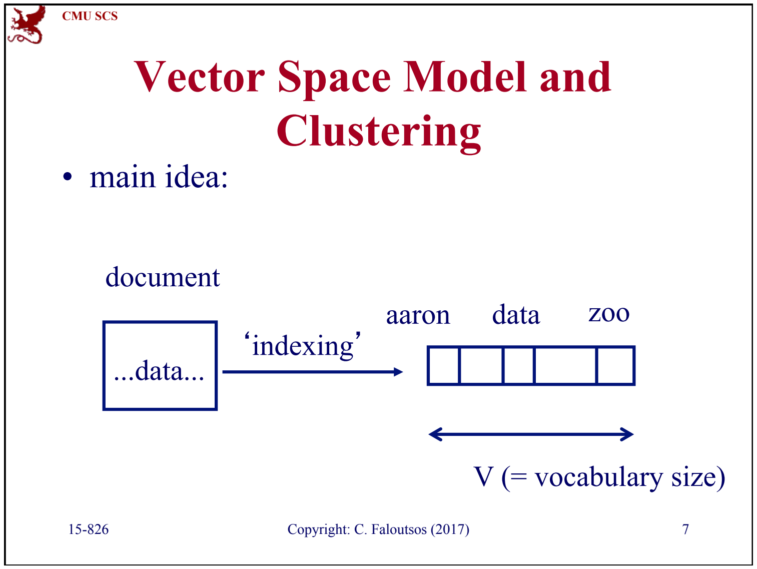
- text
  - problem
  - full text scanning
  - inversion
  - signature files
  - ➡ clustering
  - information filtering and LSI



CMU SCS

Vector Space Model and Clustering

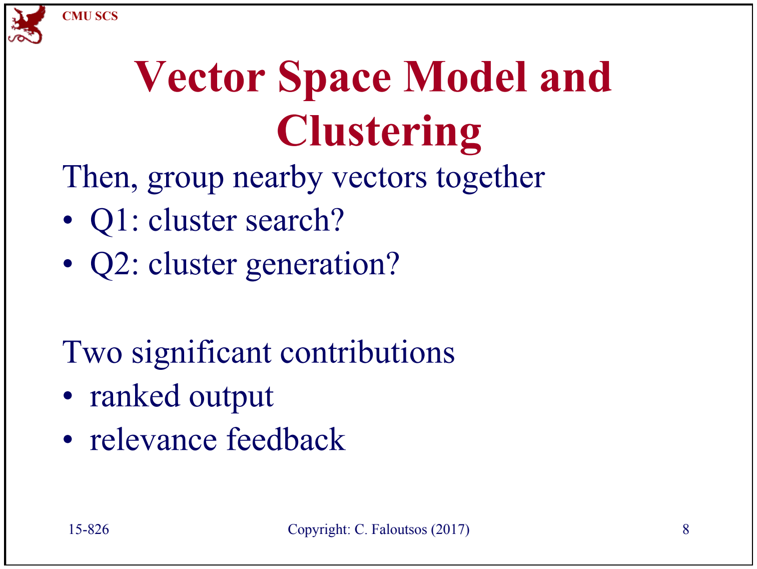
- keyword queries (vs Boolean)
- each document: -> vector (HOW?)
- each query: -> vector
- search for ‘similar’ vectors



CMU SCS

Vector Space Model and Clustering

- main idea:



CMU SCS

Vector Space Model and Clustering

Then, group nearby vectors together

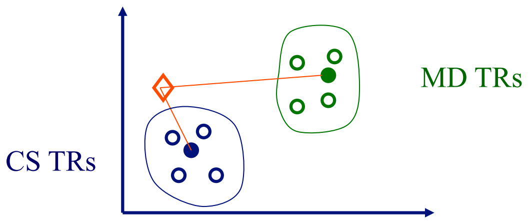
- Q1: cluster search?
- Q2: cluster generation?

Two significant contributions

- ranked output
- relevance feedback

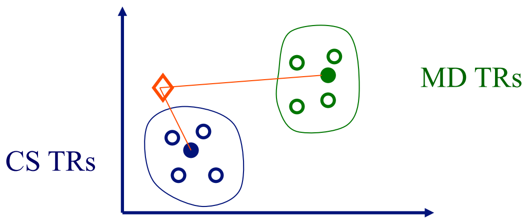
Vector Space Model and Clustering

- cluster search: visit the (k) closest superclusters; continue recursively



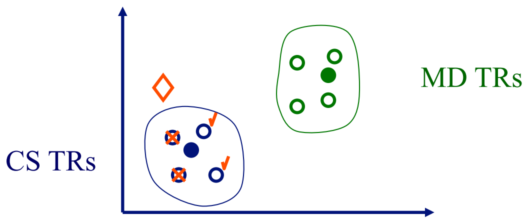
Vector Space Model and Clustering

- ranked output: easy!



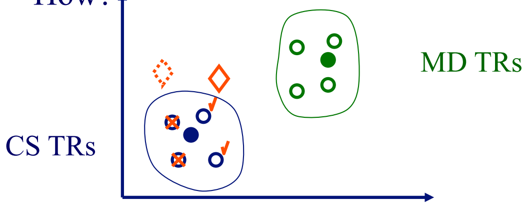
Vector Space Model and Clustering

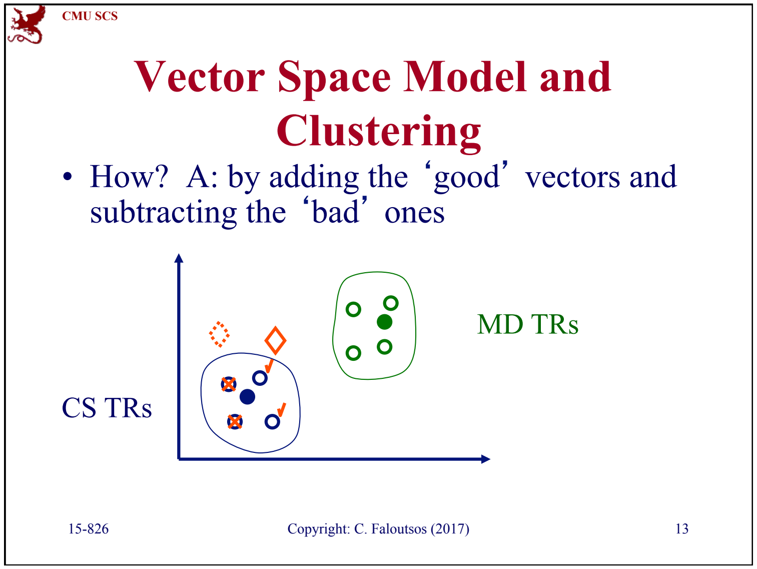
- relevance feedback (brilliant idea) [Roccio’ 73]



Vector Space Model and Clustering

- relevance feedback (brilliant idea) [Roccio’ 73]
- How?



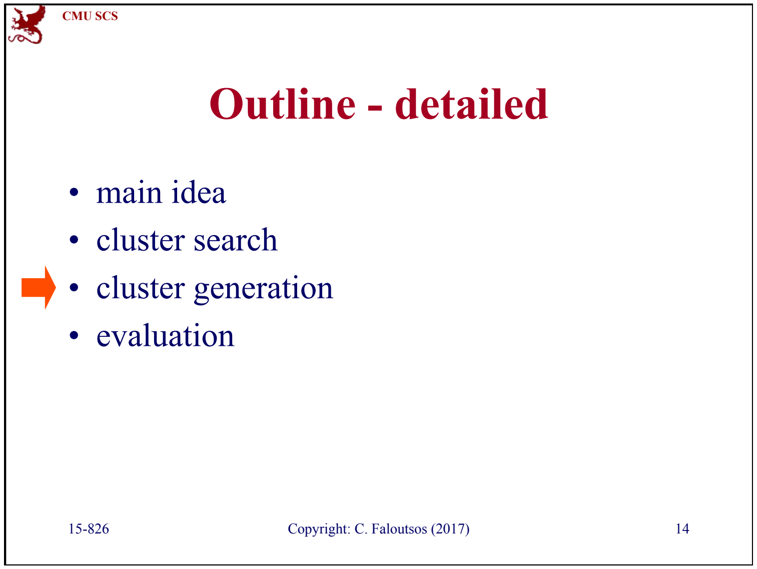


CMU SCS

Vector Space Model and Clustering

- How? A: by adding the ‘good’ vectors and subtracting the ‘bad’ ones

15-826 Copyright: C. Faloutsos (2017) 13

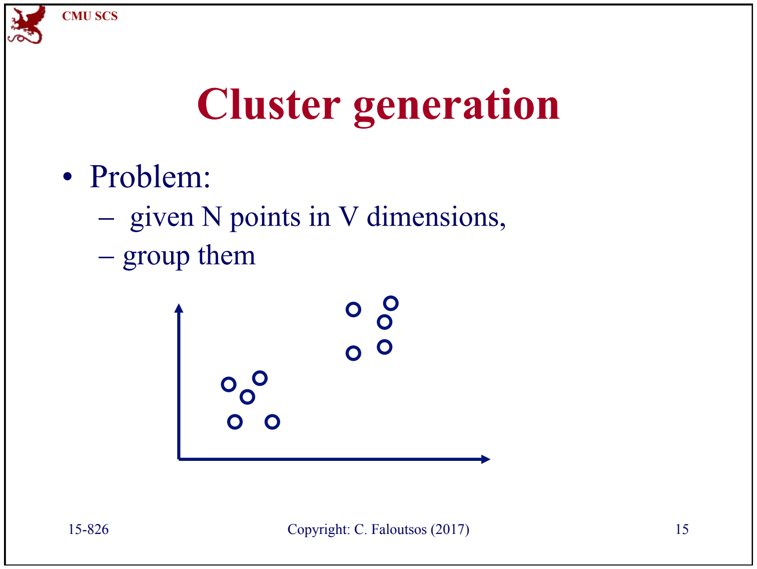


CMU SCS

Outline - detailed

- main idea
- cluster search
- cluster generation
- evaluation

15-826 Copyright: C. Faloutsos (2017) 14

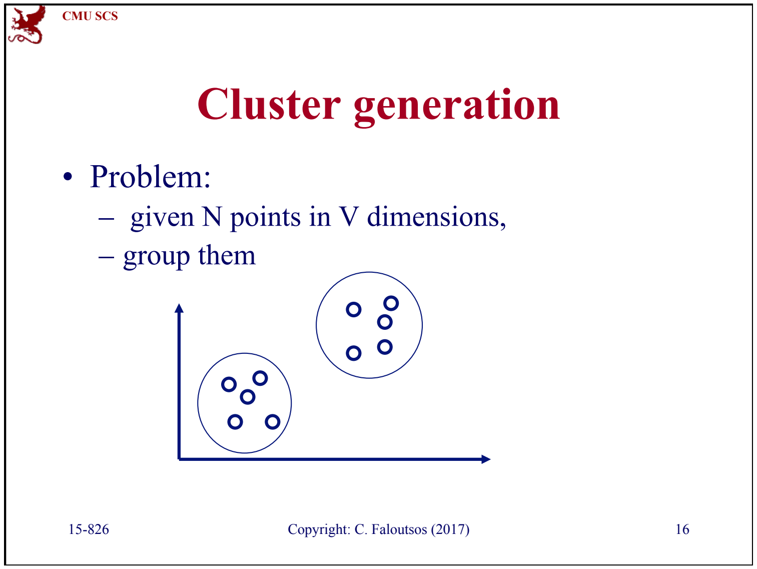


CMU SCS

Cluster generation

- Problem:
  - given N points in V dimensions,
  - group them

15-826 Copyright: C. Faloutsos (2017) 15



CMU SCS

Cluster generation

- Problem:
  - given N points in V dimensions,
  - group them

15-826 Copyright: C. Faloutsos (2017) 16

Cluster generation

We need

- Q1: document-to-document similarity
- Q2: document-to-cluster similarity

Cluster generation

Q1: document-to-document similarity
(recall: ‘bag of words’ representation)

- D1: { ‘data’, ‘retrieval’, ‘system’ }
- D2: { ‘lung’, ‘pulmonary’, ‘system’ }
- distance/similarity functions?

Cluster generation

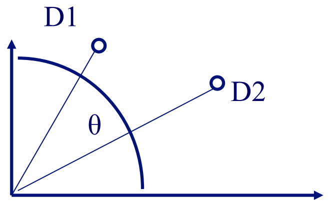
A1: # of words in common
A2: .... normalized by the vocabulary sizes
A3: .... etc

About the same performance - prevailing one:
cosine similarity

Cluster generation

cosine similarity:

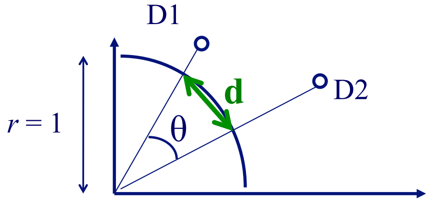
$$\text{similarity}(D1, D2) = \cos(\theta) =$$
$$\text{sum}(v_{1,i} * v_{2,i}) / \text{len}(v_1)/ \text{len}(v_2)$$



Cluster generation

cosine similarity - observations:

- related to the Euclidean distance
- weights $v_{i,j}$: according to tf/idf

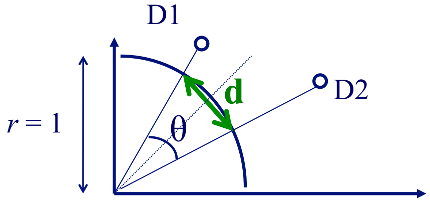


DETAILS

Cluster generation

cosine similarity - observations:

- related to the Euclidean distance
- weights $v_{i,j}$: according to tf/idf



$d = 2 * \sin(\theta/2)$

$d^2 = 2 * (1 - \cos(\theta))$

Cluster generation

tf ('term frequency')

high, if the term appears very often in this document.

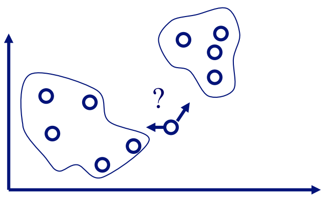
idf ('inverse document frequency')

penalizes 'common' words, that appear in almost every document

Cluster generation

We need

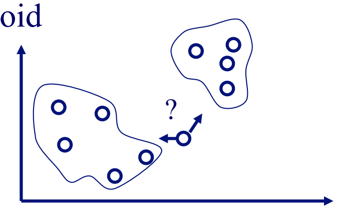
- Q1: document-to-document similarity
- ➡ Q2: document-to-cluster similarity



CMU SCS

Cluster generation

- A1: min distance (‘single-link’)
- A2: max distance (‘all-link’)
- A3: avg distance
- A4: distance to centroid



CMU SCS

Cluster generation

- A1: min distance (‘single-link’)
  - leads to elongated clusters
- A2: max distance (‘all-link’)
  - many, small, tight clusters
- A3: avg distance
  - in between the above
- A4: distance to centroid
  - fast to compute

CMU SCS

Cluster generation

We have

- document-to-document similarity
- document-to-cluster similarity

Q: How to group documents into ‘natural’ clusters

CMU SCS

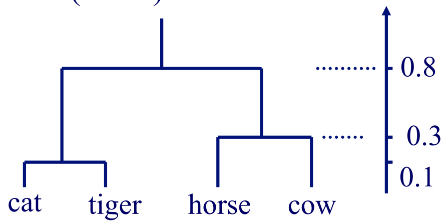
Cluster generation

A: *many-many* algorithms - in two groups [VanRijsbergen]:

- theoretically sound (O(N^2))
  - independent of the insertion order
- iterative (O(N), O(N log(N))

Cluster generation - 'sound' methods

- Approach#1: dendrograms - create a hierarchy (bottom up or top-down) - choose a cut-off (how?) and cut

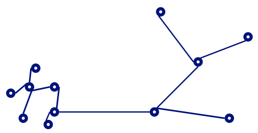


Cluster generation - 'sound' methods

- Approach#2: min. some statistical criterion (eg., sum of squares from cluster centers)
  - like 'k-means'
  - but how to decide 'k' ?

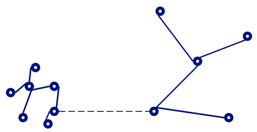
Cluster generation - 'sound' methods

- Approach#3: Graph theoretic [Zahn]:
  - build MST;
  - delete edges longer than 3* std of the local average

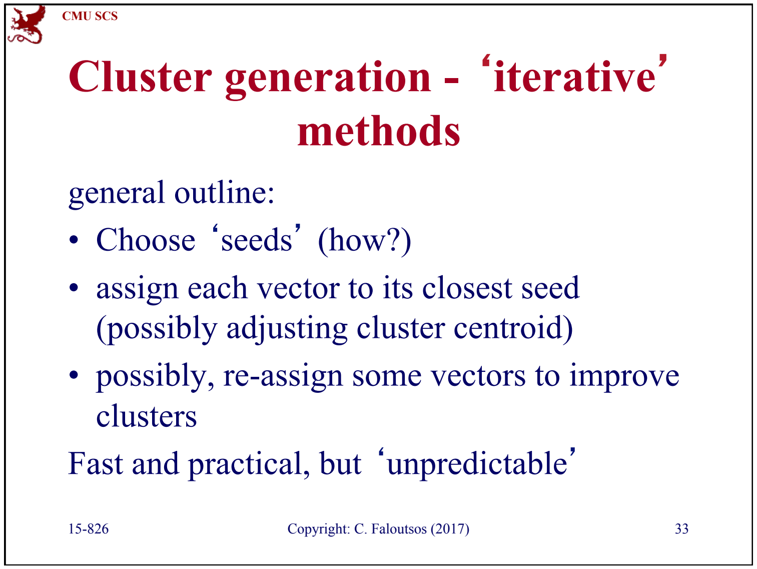


Cluster generation - 'sound' methods

- Result:



- why '3' ?
- variations
- Complexity?



CMU SCS

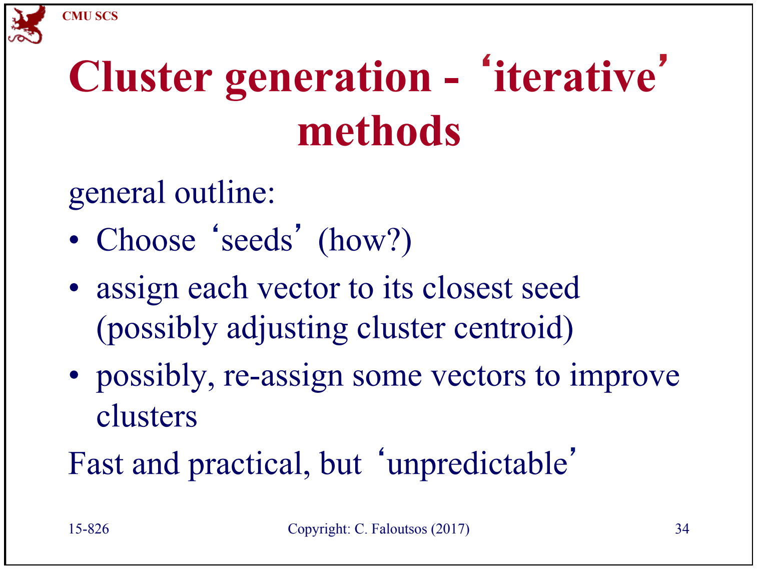
Cluster generation - ‘iterative’ methods

general outline:

- Choose ‘seeds’ (how?)
- assign each vector to its closest seed (possibly adjusting cluster centroid)
- possibly, re-assign some vectors to improve clusters

Fast and practical, but ‘unpredictable’

15-826 Copyright: C. Faloutsos (2017) 33



CMU SCS

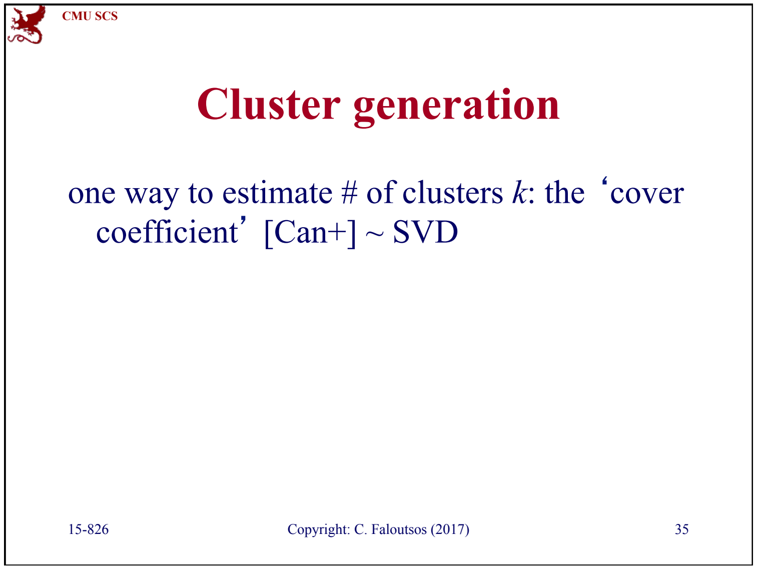
Cluster generation - ‘iterative’ methods

general outline:

- Choose ‘seeds’ (how?)
- assign each vector to its closest seed (possibly adjusting cluster centroid)
- possibly, re-assign some vectors to improve clusters

Fast and practical, but ‘unpredictable’

15-826 Copyright: C. Faloutsos (2017) 34

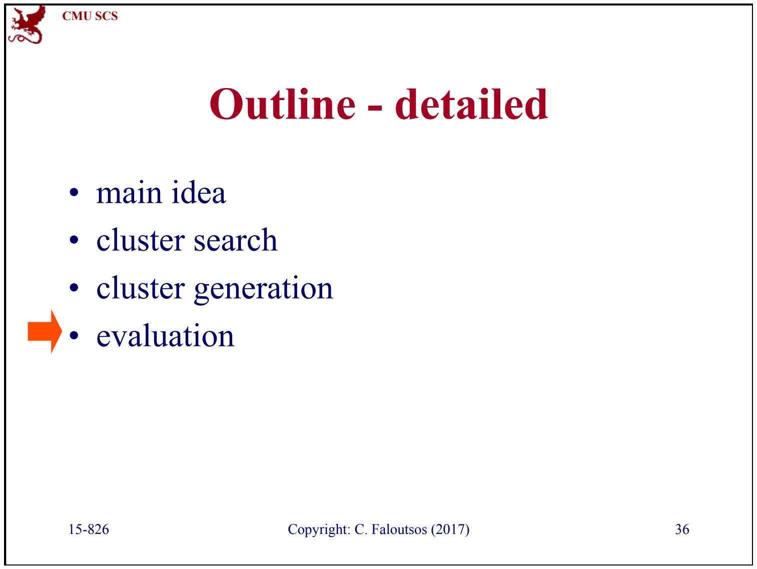


CMU SCS

Cluster generation

one way to estimate # of clusters *k*: the ‘cover coefficient’ [Can+] ~ SVD

15-826 Copyright: C. Faloutsos (2017) 35



CMU SCS

Outline - detailed

- main idea
- cluster search
- cluster generation
- ➡ evaluation

15-826 Copyright: C. Faloutsos (2017) 36

CMU SCS

Evaluation

- Q: how to measure ‘goodness’ of one distance function vs another?
- A: ground truth (by humans) and
  - ‘precision’ and ‘recall’

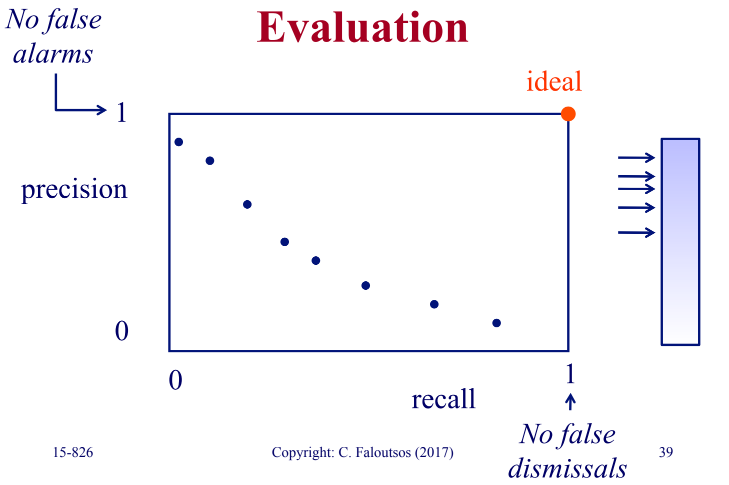
CMU SCS

Evaluation

- precision = (retrieved & relevant) / retrieved
  - 100% precision -> no false alarms
- recall = (retrieved & relevant)/ relevant
  - 100% recall -> no false dismissals

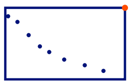
CMU SCS

Evaluation



CMU SCS

Evaluation

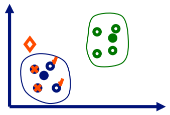


- compressing such a curve into a single number:
  - 11-point average precision
  - etc

CMU SCS

Conclusions – main ideas

- ‘bag of words’ idea + keyword queries
- Cosine similarity
- Ranked output
- Relevance feedback



CMU SCS

References

- *Modern Information Retrieval* R. Baeza-Yates, Acm Press, Berthier Ribeiro-Neto, February 1999
- Can, F. and E. A. Ozkarahan (Dec. 1990). "Concepts and Effectiveness of the Cover-Coefficient-Based Clustering Methodology for Text Databases." ACM TODS 15(4): 483-517.
- Noreault, T., M. McGill, et al. (1983). A Performance Evaluation of Similarity Measures, Document Term Weighting Schemes and Representation in a Boolean Environment. Information Retrieval Research, Butterworths.

CMU SCS

References

- Rocchio, J. J. (1971). Relevance Feedback in Information Retrieval. The SMART Retrieval System - Experiments in Automatic Document Processing. G. Salton. Englewood Cliffs, New Jersey, Prentice-Hall Inc.
- Salton, G. (1971). The SMART Retrieval System - Experiments in Automatic Document Processing. Englewood Cliffs, New Jersey, Prentice-Hall Inc.

CMU SCS

References

- Salton, G. and M. J. McGill (1983). Introduction to Modern Information Retrieval, McGraw-Hill.
- Van-Rijsbergen, C. J. (1979). Information Retrieval. London, England, Butterworths.
- Zahn, C. T. (Jan. 1971). "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters." IEEE Trans. on Computers C-20(1): 68-86.