Introduction to Information Retrieval

http://informationretrieval.org

IIR 16: Flat Clustering

Hinrich Schütze

Center for Information and Language Processing, University of Munich

2014-06-11

Overview

Outline

1. Recap
2. Clustering: Introduction
3. Clustering in IR
4. K-means
5. Evaluation
6. How many clusters?

Learning to rank for zone scoring

Given query q and document d, weighted zone scoring assigns to the pair (q, d) a score in the interval [0,1] by computing a linear combination of document zone scores, where each zone contributes a value.

- Consider a set of documents, which have l zones
- Let $g_1, ..., g_l \in [0, 1]$, such that $\sum_{i=1}^{l} g_i = 1$
- For $1 \leq i \leq l$, let s_i be the Boolean score denoting a match (or non-match) between q and the i^{th} zone
    - $s_i = 1$ if a query term occurs in zone i, 0 otherwise

Weighted zone scoring aka ranked Boolean retrieval

Rank documents according to $\sum_{i=1}^{l} g_i s_i$

Learning to rank approach: learn the weights g_i from training data

Training set for learning to rank

Φ_j	d_j	q_j	s_T	s_B	$r(d_j, q_j)$
Φ_1	37	linux	1	1	Relevant
Φ_2	37	penguin	0	1	Nonrelevant
Φ_3	238	system	0	1	Relevant
Φ_4	238	penguin	0	0	Nonrelevant
Φ_5	1741	kernel	1	1	Relevant
Φ_6	2094	driver	0	1	Relevant
Φ_7	3194	driver	1	0	Nonrelevant

Summary of learning to rank approach

- The problem of making a binary relevant/nonrelevant judgment is cast as a classification or regression problem, based on a training set of query-document pairs and associated relevance judgments.
- In principle, any method learning a classifier (including least squares regression) can be used to find this line.
- Big advantage of learning to rank: we can avoid hand-tuning scoring functions and simply learn them from training data.
- Bottleneck of learning to rank: the cost of maintaining a representative set of training examples whose relevance assessments must be made by humans.

LTR features used by Microsoft Research (1)

- Zones: body, anchor, title, url, whole document
- Features derived from standard IR models: query term number, query term ratio, length, idf, sum of term frequency, min of term frequency, max of term frequency, mean of term frequency, variance of term frequency, sum of length normalized term frequency, min of length normalized term frequency, max of length normalized term frequency, mean of length normalized term frequency, variance of length normalized term frequency, sum of tf-idf, min of tf-idf, max of tf-idf, mean of tf-idf, variance of tf-idf, boolean model, BM25

LTR features used by Microsoft Research (2)

- Language model features: LMIR.ABS, LMIR.DIR, LMIR.JM
- Web-specific features: number of slashes in url, length of url, inlink number, outlink number, PageRank, SiteRank
- Spam features: QualityScore
- Usage-based features: query-url click count, url click count, url dwell time

Ranking SVMs

- Vector of feature differences: $\Phi(d_i, d_j, q) = \psi(d_i, q) - \psi(d_j, q)$
- By hypothesis, one of d_i and d_j has been judged more relevant.
- Notation: We write $d_i \prec d_j$ for "d_i precedes d_j in the results ordering".
- If d_i is judged more relevant than d_j, then we will assign the vector $\Phi(d_i, d_j, q)$ the class $y_{ijq} = +1$; otherwise -1.
- This gives us a training set of pairs of vectors and "precedence indicators". Each of the vectors is computed as the difference of two document-query vectors.
- We can then train an SVM on this training set with the goal of obtaining a classifier that returns

$$\vec{w}^{\mathsf{T}}\Phi(d_i, d_j, q) > 0 \quad \text{iff} \quad d_i \prec d_j$$

Take-away today

Take-away today

- What is clustering?

Take-away today

- What is clustering?
- Applications of clustering in information retrieval

Take-away today

- What is clustering?
- Applications of clustering in information retrieval
- K-means algorithm

Take-away today

- What is clustering?
- Applications of clustering in information retrieval
- K-means algorithm
- Evaluation of clustering

Take-away today

- What is clustering?
- Applications of clustering in information retrieval
- K-means algorithm
- Evaluation of clustering
- How many clusters?

Outline

1. Recap
2. Clustering: Introduction
3. Clustering in IR
4. K-means
5. Evaluation
6. How many clusters?

Clustering: Definition

Clustering: Definition

- (Document) clustering is the process of grouping a set of documents into clusters of similar documents.

Clustering: Definition

- (Document) clustering is the process of grouping a set of documents into clusters of similar documents.
- Documents within a cluster should be similar.

Clustering: Definition

- (Document) clustering is the process of grouping a set of documents into clusters of similar documents.
- Documents within a cluster should be similar.
- Documents from different clusters should be dissimilar.

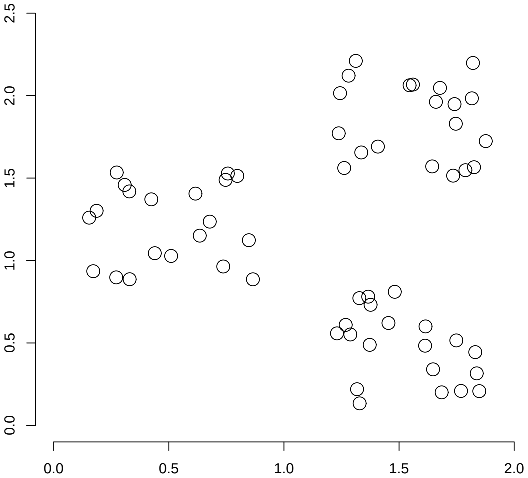
Clustering: Definition

- (Document) clustering is the process of grouping a set of documents into clusters of similar documents.
- Documents within a cluster should be similar.
- Documents from different clusters should be dissimilar.
- Clustering is the most common form of unsupervised learning.

Clustering: Definition

- (Document) clustering is the process of grouping a set of documents into clusters of similar documents.
- Documents within a cluster should be similar.
- Documents from different clusters should be dissimilar.
- Clustering is the most common form of unsupervised learning.
- Unsupervised = there are no labeled or annotated data. □

Data set with clear cluster structure



Classification vs. Clustering

Classification vs. Clustering

- Classification: supervised learning

Classification vs. Clustering

- Classification: supervised learning
- Clustering: unsupervised learning

Classification vs. Clustering

- Classification: supervised learning
- Clustering: unsupervised learning
- Classification: Classes are human-defined and part of the input to the learning algorithm.

Classification vs. Clustering

- Classification: supervised learning
- Clustering: unsupervised learning
- Classification: Classes are human-defined and part of the input to the learning algorithm.
- Clustering: Clusters are inferred from the data without human input.

Classification vs. Clustering

- Classification: supervised learning
- Clustering: unsupervised learning
- Classification: Classes are human-defined and part of the input to the learning algorithm.
- Clustering: Clusters are inferred from the data without human input.
  - However, there are many ways of influencing the outcome of clustering: number of clusters, similarity measure, representation of documents, . . . □

Outline

The cluster hypothesis

The cluster hypothesis

Cluster hypothesis. Documents in the same cluster behave similarly with respect to relevance to information needs.

The cluster hypothesis

Cluster hypothesis. Documents in the same cluster behave similarly with respect to relevance to information needs.

All applications of clustering in IR are based (directly or indirectly) on the cluster hypothesis.

The cluster hypothesis

Cluster hypothesis. Documents in the same cluster behave similarly with respect to relevance to information needs.

All applications of clustering in IR are based (directly or indirectly) on the cluster hypothesis.

Van Rijsbergen's original wording (1979): "closely associated documents tend to be relevant to the same requests".

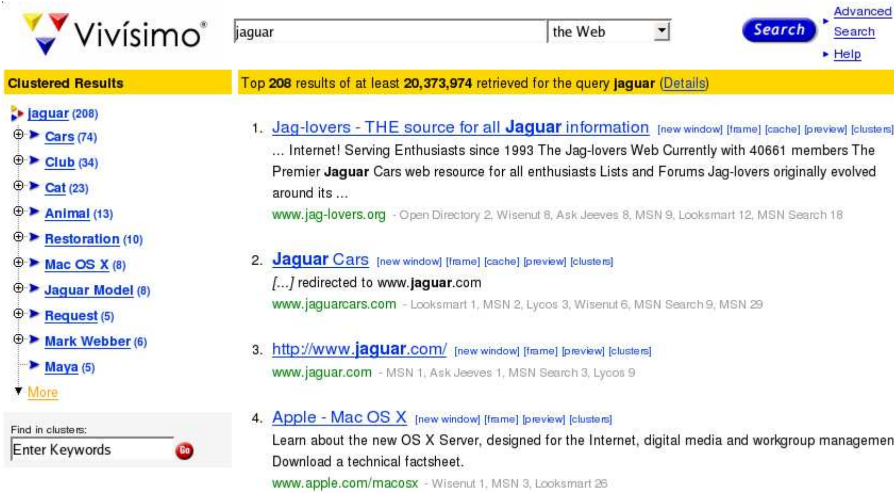
Applications of clustering in IR

Applications of clustering in IR

application	what is clustered?	benefit
search result clustering	search results	more effective information presentation to user
Scatter-Gather	(subsets of) collection	alternative user interface: "search without typing"
collection clustering	collection	effective information presentation for exploratory browsing
cluster-based retrieval	collection	higher efficiency: faster search

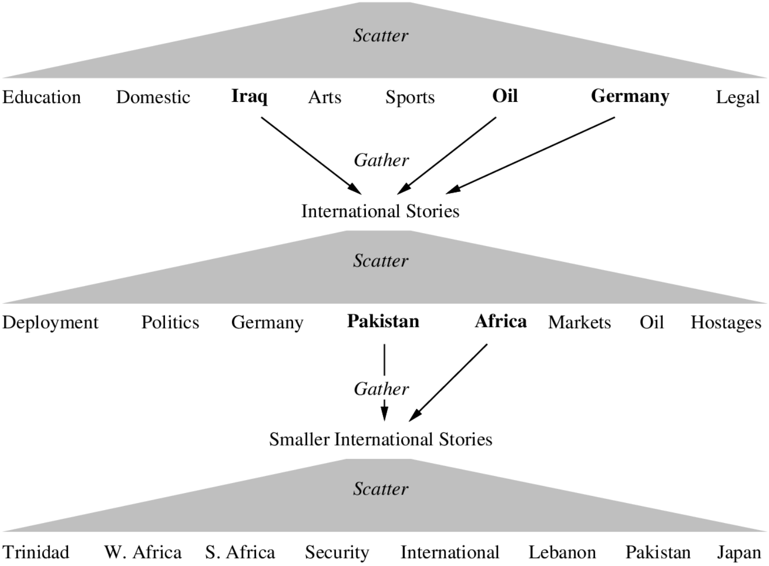
Search result clustering for better navigation

Search result clustering for better navigation



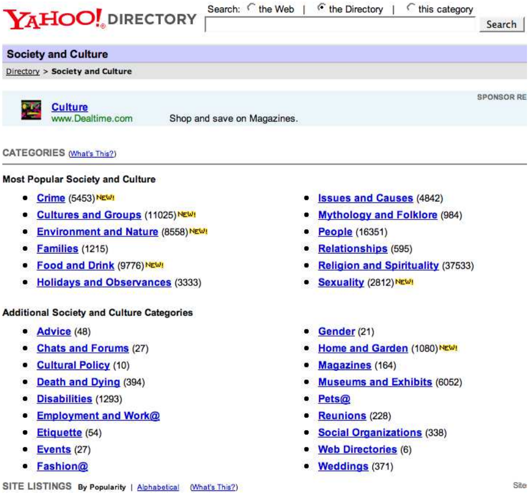
Scatter-Gather

Scatter-Gather



Global navigation: Yahoo

Global navigation: Yahoo



Global navigation: MESH (upper level)

Global navigation: MESH (upper level)

MeSH Tree Structures - 2008

Return to Entry Page

1. ⊞ Anatomy [A]
2. ⊞ Organisms [B]
3. ⊟ Diseases [C]
   - Bacterial Infections and Mycoses [C01] +
   - Virus Diseases [C02] +
   - Parasitic Diseases [C03] +
   - Neoplasms [C04] +
   - Musculoskeletal Diseases [C05] +
   - Digestive System Diseases [C06] +
   - Stomatognathic Diseases [C07] +
   - Respiratory Tract Diseases [C08] +
   - Otorhinolaryngologic Diseases [C09] +
   - Nervous System Diseases [C10] +
   - Eye Diseases [C11] +
   - Male Urogenital Diseases [C12] +
   - Female Urogenital Diseases and Pregnancy Complications [C13] +
   - Cardiovascular Diseases [C14] +
   - Hemic and Lymphatic Diseases [C15] +
   - Congenital, Hereditary, and Neonatal Diseases and Abnormalities [C16] +
   - Skin and Connective Tissue Diseases [C17] +
   - Nutritional and Metabolic Diseases [C18] +
   - Endocrine System Diseases [C19] +
   - Immune System Diseases [C20] +
   - Disorders of Environmental Origin [C21] +
   - Animal Diseases [C22] +
   - Pathological Conditions, Signs and Symptoms [C23] +
4. ⊞ Chemicals and Drugs [D]
5. ⊞ Analytical, Diagnostic and Therapeutic Techniques and Equipment [E]
6. ⊞ Psychiatry and Psychology [F]
7. ⊞ Biological Sciences [G]
8. ⊞ Natural Sciences [H]
9. ⊞ Anthropology, Education, Sociology and Social Phenomena [I]
10. ⊞ Technology, Industry, Agriculture [J]
11. ⊞ Humanities [K]

Global navigation: MESH (lower level)

Global navigation: MESH (lower level)

- Neoplasms [C04]
  - Cysts [C04.182] +
  - Hamartoma [C04.445] +
  - ➤ Neoplasms by Histologic Type [C04.557]
    - Histiocytic Disorders, Malignant [C04.557.227] +
    - Leukemia [C04.557.337] +
    - Lymphatic Vessel Tumors [C04.557.375] +
    - Lymphoma [C04.557.386] +
    - Neoplasms, Complex and Mixed [C04.557.435] +
    - Neoplasms, Connective and Soft Tissue [C04.557.450] +
    - Neoplasms, Germ Cell and Embryonal [C04.557.465] +
    - Neoplasms, Glandular and Epithelial [C04.557.470] +
    - Neoplasms, Gonadal Tissue [C04.557.475] +
    - Neoplasms, Nerve Tissue [C04.557.580] +
    - Neoplasms, Plasma Cell [C04.557.595] +
    - Neoplasms, Vascular Tissue [C04.557.645] +
    - Nevi and Melanomas [C04.557.665] +
    - Odontogenic Tumors [C04.557.695] +
  - Neoplasms by Site [C04.588] +
  - Neoplasms, Experimental [C04.619] +
  - Neoplasms, Hormone-Dependent [C04.626]
  - Neoplasms, Multiple Primary [C04.651] +
  - Neoplasms, Post-Traumatic [C04.666]
  - Neoplasms, Radiation-Induced [C04.682] +
  - Neoplasms, Second Primary [C04.692]
  - Neoplastic Processes [C04.697] +
  - Neoplastic Syndromes, Hereditary [C04.700] +
  - Paraneoplastic Syndromes [C04.730] +
  - Precancerous Conditions [C04.834] +
  - Pregnancy Complications, Neoplastic [C04.850] +
  - Tumor Virus Infections [C04.925] +

Navigational hierarchies: Manual vs. automatic creation

Navigational hierarchies: Manual vs. automatic creation

- Note: Yahoo/MESH are not examples of clustering.

Navigational hierarchies: Manual vs. automatic creation

- Note: Yahoo/MESH are not examples of clustering.
- But they are well known examples for using a global hierarchy for navigation.

Navigational hierarchies: Manual vs. automatic creation

- Note: Yahoo/MESH are not examples of clustering.
- But they are well known examples for using a global hierarchy for navigation.
- Some examples for global navigation/exploration based on clustering:

Navigational hierarchies: Manual vs. automatic creation

- Note: Yahoo/MESH are not examples of clustering.
- But they are well known examples for using a global hierarchy for navigation.
- Some examples for global navigation/exploration based on clustering:
  - Cartia

Navigational hierarchies: Manual vs. automatic creation

- Note: Yahoo/MESH are not examples of clustering.
- But they are well known examples for using a global hierarchy for navigation.
- Some examples for global navigation/exploration based on clustering:
  - Cartia
  - Themescapes

Navigational hierarchies: Manual vs. automatic creation

- Note: Yahoo/MESH are not examples of clustering.
- But they are well known examples for using a global hierarchy for navigation.
- Some examples for global navigation/exploration based on clustering:
  - Cartia
  - Themescapes
  - Google News

Desiderata for clustering

- General goal: put related docs in the same cluster, put unrelated docs in different clusters.

Desiderata for clustering

- General goal: put related docs in the same cluster, put unrelated docs in different clusters.
  - We'll see different ways of formalizing this.

Desiderata for clustering

- General goal: put related docs in the same cluster, put unrelated docs in different clusters.
  - We'll see different ways of formalizing this.
- The number of clusters should be appropriate for the data set we are clustering.

Desiderata for clustering

- General goal: put related docs in the same cluster, put unrelated docs in different clusters.
  - We'll see different ways of formalizing this.
- The number of clusters should be appropriate for the data set we are clustering.
  - Initially, we will assume the number of clusters K is given.

Desiderata for clustering

- General goal: put related docs in the same cluster, put unrelated docs in different clusters.
  - We'll see different ways of formalizing this.
- The number of clusters should be appropriate for the data set we are clustering.
  - Initially, we will assume the number of clusters K is given.
  - Later: Semiautomatic methods for determining K

Desiderata for clustering

- General goal: put related docs in the same cluster, put unrelated docs in different clusters.
  - We'll see different ways of formalizing this.
- The number of clusters should be appropriate for the data set we are clustering.
  - Initially, we will assume the number of clusters K is given.
  - Later: Semiautomatic methods for determining K
- Secondary goals in clustering

Desiderata for clustering

- General goal: put related docs in the same cluster, put unrelated docs in different clusters.
  - We'll see different ways of formalizing this.
- The number of clusters should be appropriate for the data set we are clustering.
  - Initially, we will assume the number of clusters K is given.
  - Later: Semiautomatic methods for determining K
- Secondary goals in clustering
  - Avoid very small and very large clusters

Desiderata for clustering

- General goal: put related docs in the same cluster, put unrelated docs in different clusters.
  - We'll see different ways of formalizing this.
- The number of clusters should be appropriate for the data set we are clustering.
  - Initially, we will assume the number of clusters K is given.
  - Later: Semiautomatic methods for determining K
- Secondary goals in clustering
  - Avoid very small and very large clusters
  - Define clusters that are easy to explain to the user

Desiderata for clustering

- General goal: put related docs in the same cluster, put unrelated docs in different clusters.
  - We'll see different ways of formalizing this.
- The number of clusters should be appropriate for the data set we are clustering.
  - Initially, we will assume the number of clusters K is given.
  - Later: Semiautomatic methods for determining K
- Secondary goals in clustering
  - Avoid very small and very large clusters
  - Define clusters that are easy to explain to the user
  - Many others . . .

Flat vs. Hierarchical clustering

Flat vs. Hierarchical clustering

- Flat algorithms

Flat vs. Hierarchical clustering

- Flat algorithms
  - Usually start with a random (partial) partitioning of docs into groups

Flat vs. Hierarchical clustering

- Flat algorithms
  - Usually start with a random (partial) partitioning of docs into groups
  - Refine iteratively

Flat vs. Hierarchical clustering

- Flat algorithms
  - Usually start with a random (partial) partitioning of docs into groups
  - Refine iteratively
  - Main algorithm: K-means

Flat vs. Hierarchical clustering

- Flat algorithms
  - Usually start with a random (partial) partitioning of docs into groups
  - Refine iteratively
  - Main algorithm: K-means
- Hierarchical algorithms

Flat vs. Hierarchical clustering

- Flat algorithms
  - Usually start with a random (partial) partitioning of docs into groups
  - Refine iteratively
  - Main algorithm: K-means
- Hierarchical algorithms
  - Create a hierarchy

Flat vs. Hierarchical clustering

- Flat algorithms
  - Usually start with a random (partial) partitioning of docs into groups
  - Refine iteratively
  - Main algorithm: K-means
- Hierarchical algorithms
  - Create a hierarchy
  - Bottom-up, agglomerative

Flat vs. Hierarchical clustering

- Flat algorithms
  - Usually start with a random (partial) partitioning of docs into groups
  - Refine iteratively
  - Main algorithm: K-means
- Hierarchical algorithms
  - Create a hierarchy
  - Bottom-up, agglomerative
  - Top-down, divisive

Hard vs. Soft clustering

Hard vs. Soft clustering

- Hard clustering: Each document belongs to exactly one cluster.

Hard vs. Soft clustering

- Hard clustering: Each document belongs to exactly one cluster.
  - More common and easier to do

Hard vs. Soft clustering

- Hard clustering: Each document belongs to exactly one cluster.
    - More common and easier to do
- Soft clustering: A document can belong to more than one cluster.

Hard vs. Soft clustering

- Hard clustering: Each document belongs to exactly one cluster.
  - More common and easier to do
- Soft clustering: A document can belong to more than one cluster.
  - Makes more sense for applications like creating browsable hierarchies

Hard vs. Soft clustering

- Hard clustering: Each document belongs to exactly one cluster.
  - More common and easier to do
- Soft clustering: A document can belong to more than one cluster.
  - Makes more sense for applications like creating browsable hierarchies
  - You may want to put *sneakers* in two clusters:

Hard vs. Soft clustering

- Hard clustering: Each document belongs to exactly one cluster.
  - More common and easier to do
- Soft clustering: A document can belong to more than one cluster.
  - Makes more sense for applications like creating browsable hierarchies
  - You may want to put *sneakers* in two clusters:
    - sports apparel

Hard vs. Soft clustering

- Hard clustering: Each document belongs to exactly one cluster.
  - More common and easier to do
- Soft clustering: A document can belong to more than one cluster.
  - Makes more sense for applications like creating browsable hierarchies
  - You may want to put *sneakers* in two clusters:
    - sports apparel
    - shoes

Hard vs. Soft clustering

- Hard clustering: Each document belongs to exactly one cluster.
  - More common and easier to do
- Soft clustering: A document can belong to more than one cluster.
  - Makes more sense for applications like creating browsable hierarchies
  - You may want to put *sneakers* in two clusters:
    - sports apparel
    - shoes
  - You can only do that with a soft clustering approach.

Hard vs. Soft clustering

- Hard clustering: Each document belongs to exactly one cluster.
  - More common and easier to do
- Soft clustering: A document can belong to more than one cluster.
  - Makes more sense for applications like creating browsable hierarchies
  - You may want to put *sneakers* in two clusters:
    - sports apparel
    - shoes
  - You can only do that with a soft clustering approach.
- This class: flat, hard clustering

Hard vs. Soft clustering

- Hard clustering: Each document belongs to exactly one cluster.
  - More common and easier to do
- Soft clustering: A document can belong to more than one cluster.
  - Makes more sense for applications like creating browsable hierarchies
  - You may want to put *sneakers* in two clusters:
    - sports apparel
    - shoes
  - You can only do that with a soft clustering approach.
- This class: flat, hard clustering
- Next time: hierarchical, hard clustering

Hard vs. Soft clustering

- Hard clustering: Each document belongs to exactly one cluster.
  - More common and easier to do
- Soft clustering: A document can belong to more than one cluster.
  - Makes more sense for applications like creating browsable hierarchies
  - You may want to put *sneakers* in two clusters:
    - sports apparel
    - shoes
  - You can only do that with a soft clustering approach.
- This class: flat, hard clustering
- Next time: hierarchical, hard clustering
- Next week: latent semantic indexing, a form of soft clustering

Flat algorithms

Flat algorithms

- Flat algorithms compute a partition of N documents into a set of K clusters.

Flat algorithms

- Flat algorithms compute a partition of N documents into a set of K clusters.
- Given: a set of documents and the number K

Flat algorithms

- Flat algorithms compute a partition of N documents into a set of K clusters.
- Given: a set of documents and the number K
- Find: a partition into K clusters that optimizes the chosen partitioning criterion

Flat algorithms

- Flat algorithms compute a partition of N documents into a set of K clusters.
- Given: a set of documents and the number K
- Find: a partition into K clusters that optimizes the chosen partitioning criterion
- Global optimization: exhaustively enumerate partitions, pick optimal one

Flat algorithms

- Flat algorithms compute a partition of N documents into a set of K clusters.
- Given: a set of documents and the number K
- Find: a partition into K clusters that optimizes the chosen partitioning criterion
- Global optimization: exhaustively enumerate partitions, pick optimal one
  - Not tractable

Flat algorithms

- Flat algorithms compute a partition of N documents into a set of K clusters.
- Given: a set of documents and the number K
- Find: a partition into K clusters that optimizes the chosen partitioning criterion
- Global optimization: exhaustively enumerate partitions, pick optimal one
  - Not tractable
- Effective heuristic method: K-means algorithm

Outline

K-means

K-means

- Perhaps the best known clustering algorithm

K-means

- Perhaps the best known clustering algorithm
- Simple, works well in many cases

K-means

- Perhaps the best known clustering algorithm
- Simple, works well in many cases
- Use as default / baseline for clustering documents

Document representations in clustering

Document representations in clustering

- Vector space model

Document representations in clustering

- Vector space model
- As in vector space classification, we measure relatedness between vectors by Euclidean distance . . .

Document representations in clustering

- Vector space model
- As in vector space classification, we measure relatedness between vectors by Euclidean distance . . .
- . . . which is almost equivalent to cosine similarity.

Document representations in clustering

- Vector space model
- As in vector space classification, we measure relatedness between vectors by Euclidean distance ...
- ...which is almost equivalent to cosine similarity.
- Almost: centroids are not length-normalized.

K-means: Basic idea

K-means: Basic idea

- Each cluster in K-means is defined by a centroid.

K-means: Basic idea

- Each cluster in *K*-means is defined by a centroid.
- Objective/partitioning criterion: minimize the average squared difference from the centroid

K-means: Basic idea

- Each cluster in K-means is defined by a centroid.
- Objective/partitioning criterion: minimize the average squared difference from the centroid
- Recall definition of centroid:

$$\vec{\mu}(\omega) = \frac{1}{|\omega|} \sum_{\vec{x} \in \omega} \vec{x}$$

where we use ω to denote a cluster.

K-means: Basic idea

- Each cluster in K-means is defined by a centroid.
- Objective/partitioning criterion: minimize the average squared difference from the centroid
- Recall definition of centroid:

$$\vec{\mu}(\omega) = \frac{1}{|\omega|} \sum_{\vec{x} \in \omega} \vec{x}$$

  where we use ω to denote a cluster.
- We try to find the minimum average squared difference by iterating two steps:

K-means: Basic idea

- Each cluster in K-means is defined by a centroid.
- Objective/partitioning criterion: minimize the average squared difference from the centroid
- Recall definition of centroid:

$$\vec{\mu}(\omega) = \frac{1}{|\omega|} \sum_{\vec{x} \in \omega} \vec{x}$$

  where we use ω to denote a cluster.
- We try to find the minimum average squared difference by iterating two steps:
  - reassignment: assign each vector to its closest centroid

K-means: Basic idea

- Each cluster in K-means is defined by a centroid.
- Objective/partitioning criterion: minimize the average squared difference from the centroid
- Recall definition of centroid:

$$\vec{\mu}(\omega) = \frac{1}{|\omega|} \sum_{\vec{x} \in \omega} \vec{x}$$

  where we use ω to denote a cluster.
- We try to find the minimum average squared difference by iterating two steps:
  - reassignment: assign each vector to its closest centroid
  - recomputation: recompute each centroid as the average of the vectors that were assigned to it in reassignment

K-means pseudocode (μ_k is centroid of ω_k)

K-means pseudocode (μ_k is centroid of ω_k)

K-MEANS$(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$

$(\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_K) \leftarrow$ SELECTRANDOMSEEDS$(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$

for $k \leftarrow 1$ **to** K

do $\vec{\mu}_k \leftarrow \vec{s}_k$

while stopping criterion has not been met

do for $k \leftarrow 1$ **to** K

  do $\omega_k \leftarrow \{\}$

  for $n \leftarrow 1$ **to** N

  do $j \leftarrow \arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$

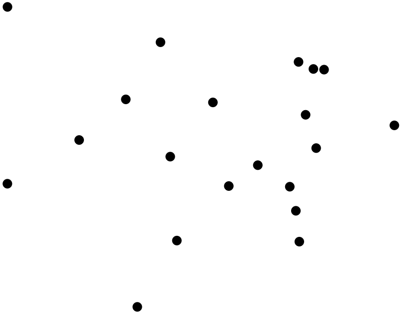
    $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ *(reassignment of vectors)*

  for $k \leftarrow 1$ **to** K

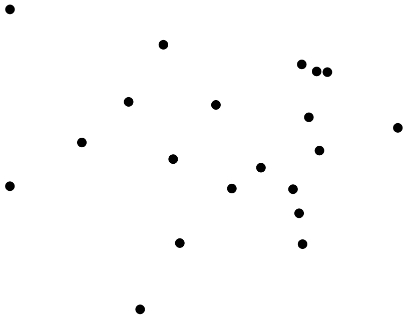
  do $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$ *(recomputation of centroids)*

return $\{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$

Worked Example: Set of points to be clustered

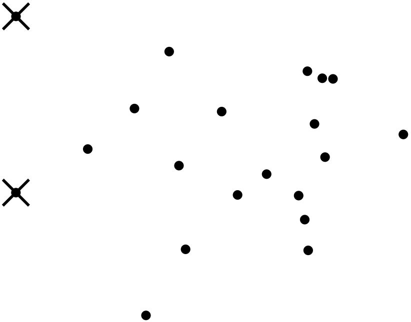


Worked Example

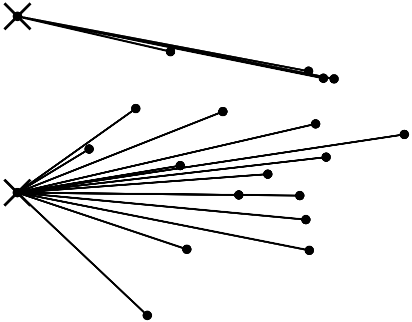


Exercise: (i) Guess what the optimal clustering into two clusters is in this case; (ii) compute the centroids of the clusters

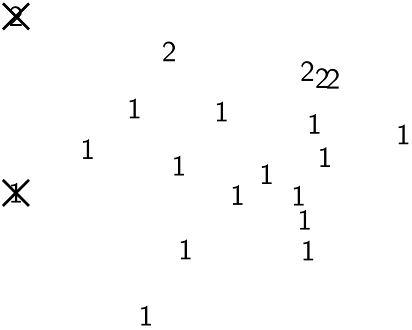
Worked Example: Random selection of initial centroids



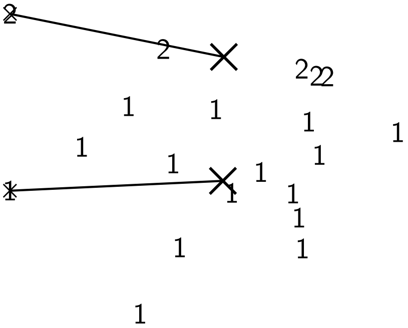
Worked Example: Assign points to closest center



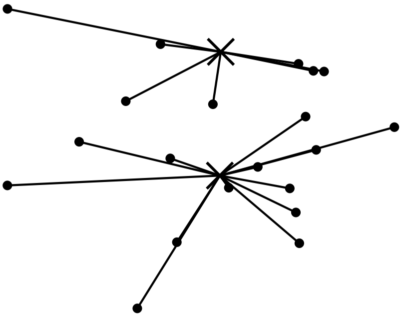
Worked Example: Assignment



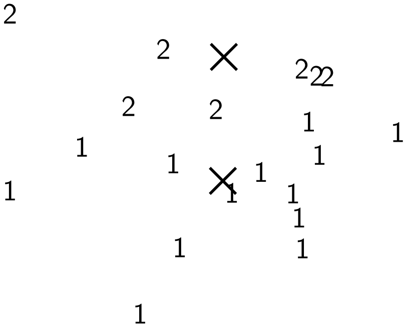
Worked Example: Recompute cluster centroids



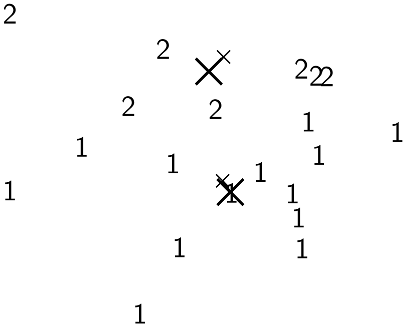
Worked Example: Assign points to closest centroid



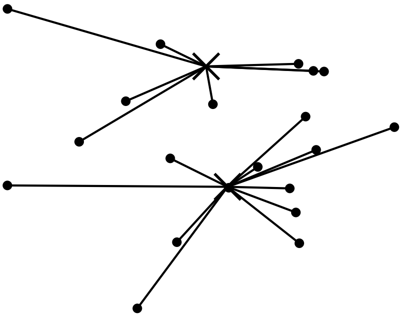
Worked Example: Assignment



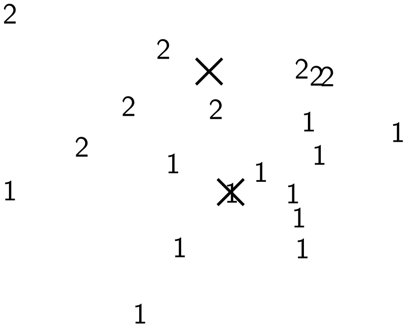
Worked Example: Recompute cluster centroids



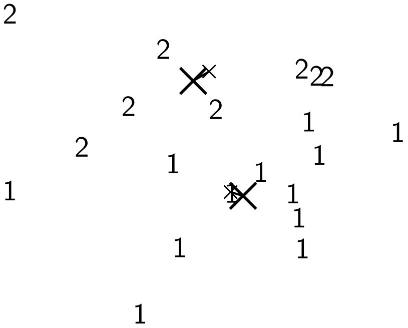
Worked Example: Assign points to closest centroid



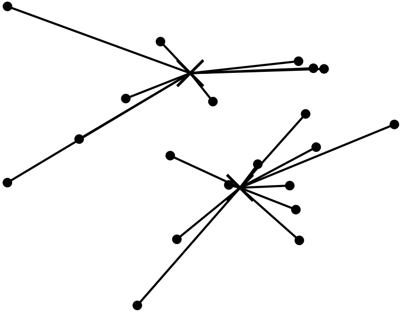
Worked Example: Assignment



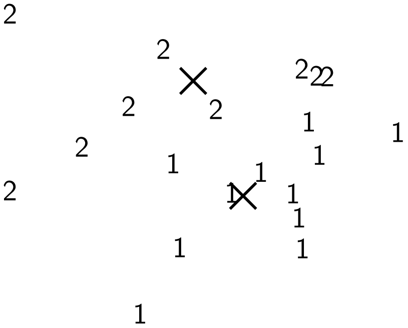
Worked Example: Recompute cluster centroids



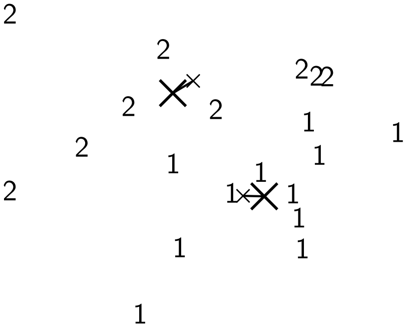
Worked Example: Assign points to closest centroid



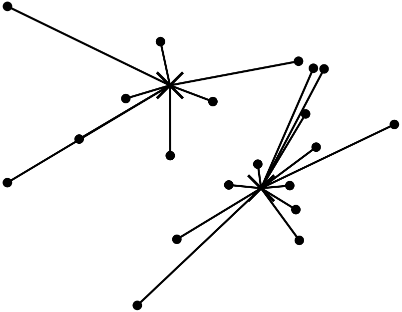
Worked Example: Assignment



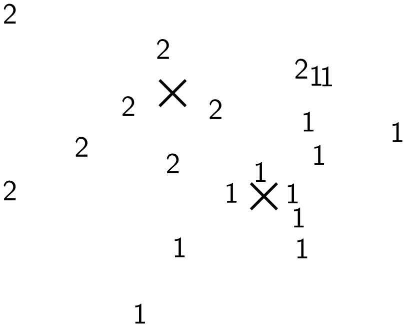
Worked Example: Recompute cluster centroids



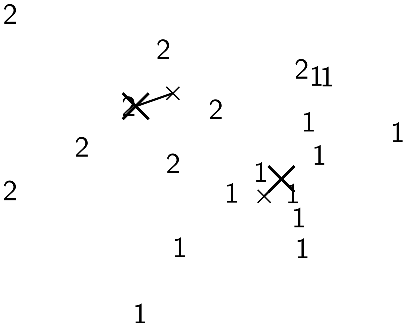
Worked Example: Assign points to closest centroid



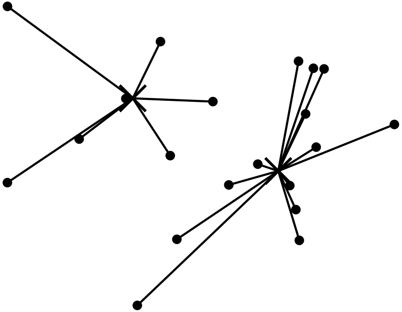
Worked Example: Assignment



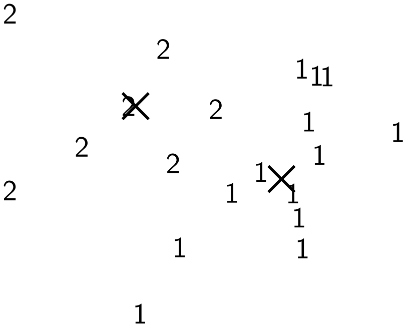
Worked Example: Recompute cluster centroids



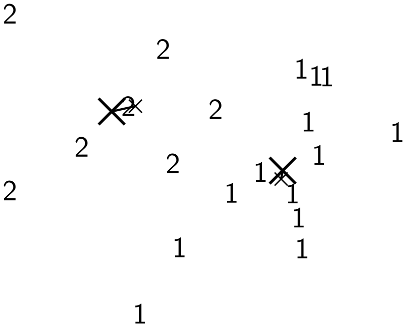
Worked Example: Assign points to closest centroid



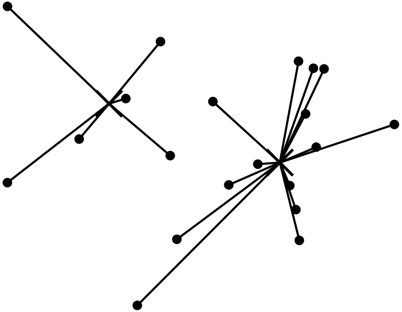
Worked Example: Assignment



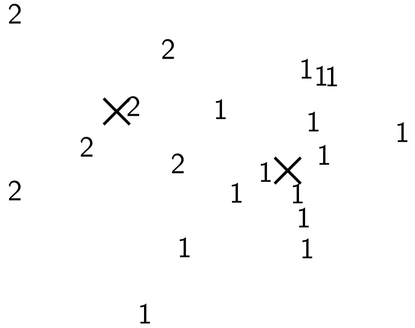
Worked Example: Recompute cluster centroids



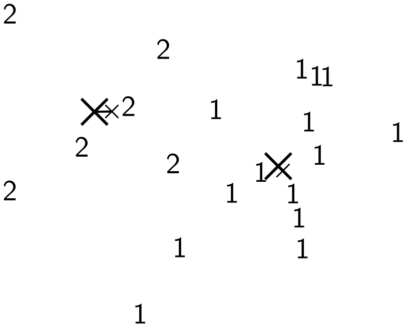
Worked Example: Assign points to closest centroid



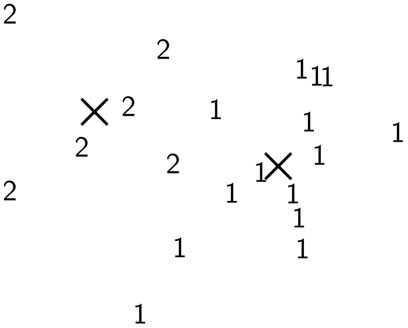
Worked Example: Assignment



Worked Example: Recompute cluster centroids



Worked Ex.: Centroids and assignments after convergence



K-means is guaranteed to converge: Proof

K-means is guaranteed to converge: Proof

- RSS = sum of all squared distances between document vector and closest centroid

K-means is guaranteed to converge: Proof

- RSS = sum of all squared distances between document vector and closest centroid
- RSS decreases during each reassignment step.

K-means is guaranteed to converge: Proof

- RSS = sum of all squared distances between document vector and closest centroid
- RSS decreases during each reassignment step.
  - because each vector is moved to a closer centroid

K-means is guaranteed to converge: Proof

- RSS = sum of all squared distances between document vector and closest centroid
- RSS decreases during each reassignment step.
  - because each vector is moved to a closer centroid
- RSS decreases during each recomputation step.

K-means is guaranteed to converge: Proof

- RSS = sum of all squared distances between document vector and closest centroid
- RSS decreases during each reassignment step.
  - because each vector is moved to a closer centroid
- RSS decreases during each recomputation step.
  - see next slide

K-means is guaranteed to converge: Proof

- RSS = sum of all squared distances between document vector and closest centroid
- RSS decreases during each reassignment step.
  - because each vector is moved to a closer centroid
- RSS decreases during each recomputation step.
  - see next slide
- There is only a finite number of clusterings.

K-means is guaranteed to converge: Proof

- RSS = sum of all squared distances between document vector and closest centroid
- RSS decreases during each reassignment step.
  - because each vector is moved to a closer centroid
- RSS decreases during each recomputation step.
  - see next slide
- There is only a finite number of clusterings.
- Thus: We must reach a fixed point.

K-means is guaranteed to converge: Proof

- RSS = sum of all squared distances between document vector and closest centroid
- RSS decreases during each reassignment step.
  - because each vector is moved to a closer centroid
- RSS decreases during each recomputation step.
  - see next slide
- There is only a finite number of clusterings.
- Thus: We must reach a fixed point.
- Assumption: Ties are broken consistently.

K-means is guaranteed to converge: Proof

- RSS = sum of all squared distances between document vector and closest centroid
- RSS decreases during each reassignment step.
  - because each vector is moved to a closer centroid
- RSS decreases during each recomputation step.
  - see next slide
- There is only a finite number of clusterings.
- Thus: We must reach a fixed point.
- Assumption: Ties are broken consistently.
- Finite set & monotonically decreasing → convergence □

Recomputation decreases average distance

Recomputation decreases average distance

$\text{RSS} = \sum_{k=1}^{K} \text{RSS}_k$ – the residual sum of squares (the "goodness" measure)

$$\text{RSS}_k(\vec{v}) = \sum_{\vec{x} \in \omega_k} \|\vec{v} - \vec{x}\|^2 = \sum_{\vec{x} \in \omega_k} \sum_{m=1}^{M} (v_m - x_m)^2$$

$$\frac{\partial \text{RSS}_k(\vec{v})}{\partial v_m} = \sum_{\vec{x} \in \omega_k} 2(v_m - x_m) = 0$$

$$v_m = \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} x_m$$

The last line is the componentwise definition of the centroid!

Recomputation decreases average distance

$\text{RSS} = \sum_{k=1}^{K} \text{RSS}_k$ – the residual sum of squares (the “goodness” measure)

$$\begin{aligned}\text{RSS}_k(\vec{v}) &= \sum_{\vec{x}\in\omega_k} \|\vec{v}-\vec{x}\|^2 = \sum_{\vec{x}\in\omega_k} \sum_{m=1}^{M} (v_m - x_m)^2 \\ \frac{\partial \text{RSS}_k(\vec{v})}{\partial v_m} &= \sum_{\vec{x}\in\omega_k} 2(v_m - x_m) = 0\end{aligned}$$

$$v_m = \frac{1}{|\omega_k|} \sum_{\vec{x}\in\omega_k} x_m$$

The last line is the componentwise definition of the centroid!
We minimize RSS_k when the old centroid is replaced with the new centroid.

Recomputation decreases average distance

$\text{RSS} = \sum_{k=1}^{K} \text{RSS}_k$ – the residual sum of squares (the "goodness" measure)

$$\begin{aligned} \text{RSS}_k(\vec{v}) &= \sum_{\vec{x} \in \omega_k} \|\vec{v} - \vec{x}\|^2 = \sum_{\vec{x} \in \omega_k} \sum_{m=1}^{M} (v_m - x_m)^2 \\ \frac{\partial \text{RSS}_k(\vec{v})}{\partial v_m} &= \sum_{\vec{x} \in \omega_k} 2(v_m - x_m) = 0 \end{aligned}$$

$$v_m = \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} x_m$$

The last line is the componentwise definition of the centroid!
We minimize RSS_k when the old centroid is replaced with the new centroid. RSS, the sum of the RSS_k, must then also decrease during recomputation. □

K-means is guaranteed to converge

K-means is guaranteed to converge

- But we don't know how long convergence will take!

K-means is guaranteed to converge

- But we don't know how long convergence will take!
- If we don't care about a few docs switching back and forth, then convergence is usually fast ($<$ 10-20 iterations).

K-means is guaranteed to converge

- But we don't know how long convergence will take!
- If we don't care about a few docs switching back and forth, then convergence is usually fast ($<$ 10-20 iterations).
- However, complete convergence can take many more iterations.

Optimality of K-means

- Convergence $\neq$ optimality

Optimality of K-means

- Convergence $\neq$ optimality
- Convergence does not mean that we converge to the optimal clustering!

Optimality of K-means

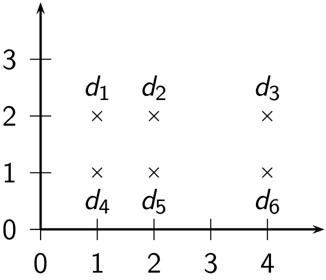
- Convergence $\neq$ optimality
- Convergence does not mean that we converge to the optimal clustering!
- This is the great weakness of K-means.

Optimality of K-means

- Convergence $\neq$ optimality
- Convergence does not mean that we converge to the optimal clustering!
- This is the great weakness of K-means.
- If we start with a bad set of seeds, the resulting clustering can be horrible. □

Exercise: Suboptimal clustering

Exercise: Suboptimal clustering



- What is the optimal clustering for $K = 2$?
- Do we converge on this clustering for arbitrary seeds d_i, d_j?

Initialization of K-means

Initialization of K-means

- Random seed selection is just one of many ways K-means can be initialized.

Initialization of K-means

- Random seed selection is just one of many ways K-means can be initialized.
- Random seed selection is not very robust: It's easy to get a suboptimal clustering.

Initialization of K-means

- Random seed selection is just one of many ways K-means can be initialized.
- Random seed selection is not very robust: It's easy to get a suboptimal clustering.
- Better ways of computing initial centroids:

Initialization of K-means

- Random seed selection is just one of many ways K-means can be initialized.
- Random seed selection is not very robust: It's easy to get a suboptimal clustering.
- Better ways of computing initial centroids:
  - Select seeds not randomly, but using some heuristic (e.g., filter out outliers or find a set of seeds that has "good coverage" of the document space)

Initialization of K-means

- Random seed selection is just one of many ways K-means can be initialized.
- Random seed selection is not very robust: It's easy to get a suboptimal clustering.
- Better ways of computing initial centroids:
  - Select seeds not randomly, but using some heuristic (e.g., filter out outliers or find a set of seeds that has "good coverage" of the document space)
  - Use hierarchical clustering to find good seeds

Initialization of K-means

- Random seed selection is just one of many ways K-means can be initialized.
- Random seed selection is not very robust: It's easy to get a suboptimal clustering.
- Better ways of computing initial centroids:
  - Select seeds not randomly, but using some heuristic (e.g., filter out outliers or find a set of seeds that has "good coverage" of the document space)
  - Use hierarchical clustering to find good seeds
  - Select i (e.g., $i = 10$) different random sets of seeds, do a K-means clustering for each, select the clustering with lowest RSS □

Time complexity of K-means

Time complexity of K-means

- Computing one distance of two vectors is $O(M)$.

Time complexity of K-means

- Computing one distance of two vectors is $O(M)$.
- Reassignment step: $O(KNM)$ (we need to compute KN document-centroid distances)

Time complexity of K-means

- Computing one distance of two vectors is $O(M)$.
- Reassignment step: $O(KNM)$ (we need to compute KN document-centroid distances)
- Recomputation step: $O(NM)$ (we need to add each of the document's $< M$ values to one of the centroids)

Time complexity of K-means

- Computing one distance of two vectors is $O(M)$.
- Reassignment step: $O(KNM)$ (we need to compute KN document-centroid distances)
- Recomputation step: $O(NM)$ (we need to add each of the document's $< M$ values to one of the centroids)
- Assume number of iterations bounded by I

Time complexity of K-means

- Computing one distance of two vectors is $O(M)$.
- Reassignment step: $O(KNM)$ (we need to compute KN document-centroid distances)
- Recomputation step: $O(NM)$ (we need to add each of the document's $< M$ values to one of the centroids)
- Assume number of iterations bounded by I
- Overall complexity: $O(IKNM)$ – linear in all important dimensions

Time complexity of K-means

- Computing one distance of two vectors is $O(M)$.
- Reassignment step: $O(KNM)$ (we need to compute KN document-centroid distances)
- Recomputation step: $O(NM)$ (we need to add each of the document's $< M$ values to one of the centroids)
- Assume number of iterations bounded by I
- Overall complexity: $O(IKNM)$ – linear in all important dimensions
- However: This is not a real worst-case analysis.

Time complexity of K-means

- Computing one distance of two vectors is $O(M)$.
- Reassignment step: $O(KNM)$ (we need to compute KN document-centroid distances)
- Recomputation step: $O(NM)$ (we need to add each of the document's $< M$ values to one of the centroids)
- Assume number of iterations bounded by I
- Overall complexity: $O(IKNM)$ – linear in all important dimensions
- However: This is not a real worst-case analysis.
- In pathological cases, complexity can be worse than linear. □

Outline

1. Recap
2. Clustering: Introduction
3. Clustering in IR
4. K-means
5. Evaluation
6. How many clusters?

What is a good clustering?

What is a good clustering?

- Internal criteria

What is a good clustering?

- Internal criteria
  - Example of an internal criterion: RSS in K-means

What is a good clustering?

- Internal criteria
  - Example of an internal criterion: RSS in K-means
- But an internal criterion often does not evaluate the actual utility of a clustering in the application.

What is a good clustering?

- Internal criteria
  - Example of an internal criterion: RSS in K-means
- But an internal criterion often does not evaluate the actual utility of a clustering in the application.
- Alternative: External criteria

What is a good clustering?

- Internal criteria
  - Example of an internal criterion: RSS in K-means
- But an internal criterion often does not evaluate the actual utility of a clustering in the application.
- Alternative: External criteria
  - Evaluate with respect to a human-defined classification

External criteria for clustering quality

External criteria for clustering quality

- Based on a gold standard data set, e.g., the Reuters collection we also used for the evaluation of classification

External criteria for clustering quality

- Based on a gold standard data set, e.g., the Reuters collection we also used for the evaluation of classification
- Goal: Clustering should reproduce the classes in the gold standard

External criteria for clustering quality

- Based on a gold standard data set, e.g., the Reuters collection we also used for the evaluation of classification
- Goal: Clustering should reproduce the classes in the gold standard
- (But we only want to reproduce how documents are divided into groups, not the class labels.)

External criteria for clustering quality

- Based on a gold standard data set, e.g., the Reuters collection we also used for the evaluation of classification
- Goal: Clustering should reproduce the classes in the gold standard
- (But we only want to reproduce how documents are divided into groups, not the class labels.)
- First measure for how well we were able to reproduce the classes: purity

External criterion: Purity

External criterion: Purity

$$\text{purity}(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

- $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ is the set of clusters and $C = \{c_1, c_2, \ldots, c_J\}$ is the set of classes.

External criterion: Purity

$$\text{purity}(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

- $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ is the set of clusters and $C = \{c_1, c_2, \ldots, c_J\}$ is the set of classes.
- For each cluster ω_k: find class c_j with most members n_{kj} in ω_k

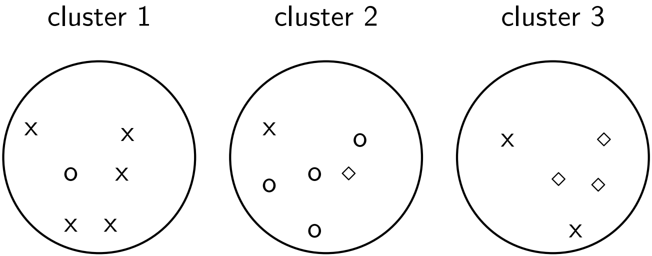
External criterion: Purity

$$\text{purity}(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

- $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ is the set of clusters and $C = \{c_1, c_2, \ldots, c_J\}$ is the set of classes.
- For each cluster ω_k: find class c_j with most members n_{kj} in ω_k
- Sum all n_{kj} and divide by total number of points □

Example for computing purity

Example for computing purity



To compute purity: $5 = \max_j |\omega_1 \cap c_j|$ (class x, cluster 1); $4 = \max_j |\omega_2 \cap c_j|$ (class o, cluster 2); and $3 = \max_j |\omega_3 \cap c_j|$ (class $\diamond$, cluster 3). Purity is $(1/17) \times (5 + 4 + 3) \approx 0.71$. □

Another external criterion: Rand index

Another external criterion: Rand index

- Purity can be increased easily by increasing K – a measure that does not have this problem: Rand index.

Another external criterion: Rand index

- Purity can be increased easily by increasing K – a measure that does not have this problem: Rand index.
- Definition: $\text{RI} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}}$

Another external criterion: Rand index

- Purity can be increased easily by increasing K – a measure that does not have this problem: Rand index.
- Definition: $\text{RI} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}}$
- Based on 2x2 contingency table of all pairs of documents:

	same cluster	different clusters
same class	true positives (TP)	false negatives (FN)
different classes	false positives (FP)	true negatives (TN)

Another external criterion: Rand index

- Purity can be increased easily by increasing K – a measure that does not have this problem: Rand index.
- Definition: $\text{RI} = \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}}$
- Based on 2x2 contingency table of all pairs of documents:

	same cluster	different clusters
same class	true positives (TP)	false negatives (FN)
different classes	false positives (FP)	true negatives (TN)

- TP+FN+FP+TN is the total number of pairs.

Another external criterion: Rand index

- Purity can be increased easily by increasing K – a measure that does not have this problem: Rand index.
- Definition: $\text{RI} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}}$
- Based on 2x2 contingency table of all pairs of documents:

	same cluster	different clusters
same class	true positives (TP)	false negatives (FN)
different classes	false positives (FP)	true negatives (TN)

- TP+FN+FP+TN is the total number of pairs.
- $\text{TP}+\text{FN}+\text{FP}+\text{TN} = \binom{N}{2}$ for N documents.

Another external criterion: Rand index

- Purity can be increased easily by increasing K – a measure that does not have this problem: Rand index.
- Definition: $\text{RI} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}}$
- Based on 2x2 contingency table of all pairs of documents:

	same cluster	different clusters
same class	true positives (TP)	false negatives (FN)
different classes	false positives (FP)	true negatives (TN)

- TP+FN+FP+TN is the total number of pairs.
- $\text{TP}+\text{FN}+\text{FP}+\text{TN} = \binom{N}{2}$ for N documents.
- Example: $\binom{17}{2} = 136$ in o/$\diamond$/x example

Another external criterion: Rand index

- Purity can be increased easily by increasing K – a measure that does not have this problem: Rand index.
- Definition: $\text{RI} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}}$
- Based on 2x2 contingency table of all pairs of documents:

	same cluster	different clusters
same class	true positives (TP)	false negatives (FN)
different classes	false positives (FP)	true negatives (TN)

- TP+FN+FP+TN is the total number of pairs.
- $\text{TP}+\text{FN}+\text{FP}+\text{TN} = \binom{N}{2}$ for N documents.
- Example: $\binom{17}{2} = 136$ in o/$\diamond$/x example
- Each pair is either positive or negative (the clustering puts the two documents in the same or in different clusters) . . .

Another external criterion: Rand index

- Purity can be increased easily by increasing K – a measure that does not have this problem: Rand index.
- Definition: $\text{RI} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}}$
- Based on 2x2 contingency table of all pairs of documents:

	same cluster	different clusters
same class	true positives (TP)	false negatives (FN)
different classes	false positives (FP)	true negatives (TN)

- TP+FN+FP+TN is the total number of pairs.
- $\text{TP}+\text{FN}+\text{FP}+\text{TN} = \binom{N}{2}$ for N documents.
- Example: $\binom{17}{2} = 136$ in o/$\diamond$/x example
- Each pair is either positive or negative (the clustering puts the two documents in the same or in different clusters) . . .
- . . . and either "true" (correct) or "false" (incorrect): the clustering decision is correct or incorrect. □

Rand Index: Example

Rand Index: Example

As an example, we compute RI for the o/⋄/x example. We first compute TP + FP. The three clusters contain 6, 6, and 5 points, respectively, so the total number of "positives" or pairs of documents that are in the same cluster is:

$$\text{TP} + \text{FP} = \binom{6}{2} + \binom{6}{2} + \binom{5}{2} = 40$$

Of these, the x pairs in cluster 1, the o pairs in cluster 2, the ⋄ pairs in cluster 3, and the x pair in cluster 3 are true positives:

$$\text{TP} = \binom{5}{2} + \binom{4}{2} + \binom{3}{2} + \binom{2}{2} = 20$$

Thus, FP $= 40 - 20 = 20$.
FN and TN are computed similarly.

Rand measure for the o/$\diamond$/x example

Rand measure for the o/$\diamond$/x example

	same cluster	different clusters
same class	TP = 20	FN = 24
different classes	FP = 20	TN = 72

RI is then $(20 + 72)/(20 + 20 + 24 + 72) \approx 0.68$.

Two other external evaluation measures

Two other external evaluation measures

- Two other measures

Two other external evaluation measures

- Two other measures
- Normalized mutual information (NMI)

Two other external evaluation measures

- Two other measures
- Normalized mutual information (NMI)
  - How much information does the clustering contain about the classification?

Two other external evaluation measures

- Two other measures
- Normalized mutual information (NMI)
  - How much information does the clustering contain about the classification?
  - Singleton clusters (number of clusters = number of docs) have maximum MI

Two other external evaluation measures

- Two other measures
- Normalized mutual information (NMI)
  - How much information does the clustering contain about the classification?
  - Singleton clusters (number of clusters = number of docs) have maximum MI
  - Therefore: normalize by entropy of clusters and classes

Two other external evaluation measures

- Two other measures
- Normalized mutual information (NMI)
  - How much information does the clustering contain about the classification?
  - Singleton clusters (number of clusters = number of docs) have maximum MI
  - Therefore: normalize by entropy of clusters and classes
- F measure

Two other external evaluation measures

- Two other measures
- Normalized mutual information (NMI)
  - How much information does the clustering contain about the classification?
  - Singleton clusters (number of clusters = number of docs) have maximum MI
  - Therefore: normalize by entropy of clusters and classes
- F measure
  - Like Rand, but "precision" and "recall" can be weighted □

Evaluation results for the o/$\diamond$/x example

Evaluation results for the o/◇/x example

	purity	NMI	RI	F_5
lower bound	0.0	0.0	0.0	0.0
maximum	1.0	1.0	1.0	1.0
value for example	0.71	0.36	0.68	0.46

All four measures range from 0 (really bad clustering) to 1 (perfect clustering). □

Outline

How many clusters?

How many clusters?

- Number of clusters K is given in many applications.

How many clusters?

- Number of clusters K is given in many applications.
  - E.g., there may be an external constraint on K. Example: In the case of Scatter-Gather, it was hard to show more than 10–20 clusters on a monitor in the 90s.

How many clusters?

- Number of clusters K is given in many applications.
  - E.g., there may be an external constraint on K. Example: In the case of Scatter-Gather, it was hard to show more than 10–20 clusters on a monitor in the 90s.
- What if there is no external constraint? Is there a “right” number of clusters?

How many clusters?

- Number of clusters K is given in many applications.
  - E.g., there may be an external constraint on K. Example: In the case of Scatter-Gather, it was hard to show more than 10–20 clusters on a monitor in the 90s.
- What if there is no external constraint? Is there a "right" number of clusters?
- One way to go: define an optimization criterion

How many clusters?

- Number of clusters K is given in many applications.
  - E.g., there may be an external constraint on K. Example: In the case of Scatter-Gather, it was hard to show more than 10–20 clusters on a monitor in the 90s.
- What if there is no external constraint? Is there a "right" number of clusters?
- One way to go: define an optimization criterion
  - Given docs, find K for which the optimum is reached.

How many clusters?

- Number of clusters K is given in many applications.
    - E.g., there may be an external constraint on K. Example: In the case of Scatter-Gather, it was hard to show more than 10–20 clusters on a monitor in the 90s.
- What if there is no external constraint? Is there a "right" number of clusters?
- One way to go: define an optimization criterion
    - Given docs, find K for which the optimum is reached.
    - What optimization criterion can we use?

How many clusters?

- Number of clusters K is given in many applications.
  - E.g., there may be an external constraint on K. Example: In the case of Scatter-Gather, it was hard to show more than 10–20 clusters on a monitor in the 90s.
- What if there is no external constraint? Is there a "right" number of clusters?
- One way to go: define an optimization criterion
  - Given docs, find K for which the optimum is reached.
  - What optimization criterion can we use?
  - We can't use RSS or average squared distance from centroid as criterion: always chooses $K = N$ clusters. □

Exercise

Exercise

- Your job is to develop the clustering algorithms for a competitor to news.google.com

Exercise

- Your job is to develop the clustering algorithms for a competitor to news.google.com
- You want to use K-means clustering.

Exercise

- Your job is to develop the clustering algorithms for a competitor to news.google.com
- You want to use K-means clustering.
- How would you determine K?

Simple objective function for K: Basic idea

Simple objective function for K: Basic idea

- Start with 1 cluster ($K = 1$)

Simple objective function for K: Basic idea

- Start with 1 cluster ($K = 1$)
- Keep adding clusters ($=$ keep increasing K)

Simple objective function for K: Basic idea

- Start with 1 cluster ($K = 1$)
- Keep adding clusters (= keep increasing K)
- Add a penalty for each new cluster

Simple objective function for K: Basic idea

- Start with 1 cluster ($K = 1$)
- Keep adding clusters (= keep increasing K)
- Add a penalty for each new cluster
- Then trade off cluster penalties against average squared distance from centroid

Simple objective function for K: Basic idea

- Start with 1 cluster ($K = 1$)
- Keep adding clusters (= keep increasing K)
- Add a penalty for each new cluster
- Then trade off cluster penalties against average squared distance from centroid
- Choose the value of K with the best tradeoff □

Simple objective function for K: Formalization

Simple objective function for K: Formalization

- Given a clustering, define the cost for a document as (squared) distance to centroid

Simple objective function for K: Formalization

- Given a clustering, define the cost for a document as (squared) distance to centroid
- Define total distortion RSS(K) as sum of all individual document costs (corresponds to average distance)

Simple objective function for K: Formalization

- Given a clustering, define the cost for a document as (squared) distance to centroid
- Define total distortion RSS(K) as sum of all individual document costs (corresponds to average distance)
- Then: penalize each cluster with a cost λ

Simple objective function for K: Formalization

- Given a clustering, define the cost for a document as (squared) distance to centroid
- Define total distortion RSS(K) as sum of all individual document costs (corresponds to average distance)
- Then: penalize each cluster with a cost λ
- Thus for a clustering with K clusters, total cluster penalty is $K\lambda$

Simple objective function for K: Formalization

- Given a clustering, define the cost for a document as (squared) distance to centroid
- Define total distortion RSS(K) as sum of all individual document costs (corresponds to average distance)
- Then: penalize each cluster with a cost λ
- Thus for a clustering with K clusters, total cluster penalty is $K\lambda$
- Define the total cost of a clustering as distortion plus total cluster penalty: $\text{RSS(K)} + K\lambda$

Simple objective function for K: Formalization

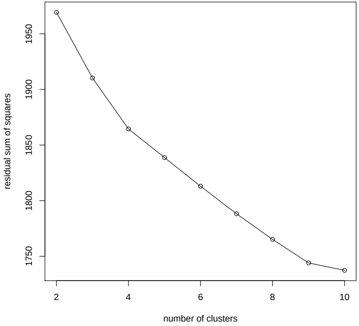
- Given a clustering, define the cost for a document as (squared) distance to centroid
- Define total distortion RSS(K) as sum of all individual document costs (corresponds to average distance)
- Then: penalize each cluster with a cost λ
- Thus for a clustering with K clusters, total cluster penalty is $K\lambda$
- Define the total cost of a clustering as distortion plus total cluster penalty: $\text{RSS(K)} + K\lambda$
- Select K that minimizes $(\text{RSS(K)} + K\lambda)$

Simple objective function for K: Formalization

- Given a clustering, define the cost for a document as (squared) distance to centroid
- Define total distortion RSS(K) as sum of all individual document costs (corresponds to average distance)
- Then: penalize each cluster with a cost λ
- Thus for a clustering with K clusters, total cluster penalty is $K\lambda$
- Define the total cost of a clustering as distortion plus total cluster penalty: RSS(K) $+ K\lambda$
- Select K that minimizes (RSS(K) $+ K\lambda$)
- Still need to determine good value for λ ...

Finding the "knee" in the curve

Finding the “knee” in the curve



Pick the number of clusters where curve “flattens”. Here: 4 or 9.

Take-away today

- What is clustering?
- Applications of clustering in information retrieval
- K-means algorithm
- Evaluation of clustering
- How many clusters?

Resources

- Chapter 16 of IIR
- Resources at `http://cislmu.org`
  - Keith van Rijsbergen on the cluster hypothesis (he was one of the originators)
  - Bing/Carrot2/Clusty: search result clustering systems
  - Stirling number: the number of distinct k-clusterings of n items