Introduction to Information Retrieval
`http://informationretrieval.org`

IIR 16: Flat Clustering

Hinrich Schütze

Institute for Natural Language Processing, Universität Stuttgart

2008.06.24

Overview

Outline

MI example for *poultry*/EXPORT in Reuters

	$e_c = e_{poultry} = 1$	$e_c = e_{poultry} = 0$
$e_t = e_{\text{EXPORT}} = 1$	$N_{11} = 49$	$N_{10} = 27{,}652$
$e_t = e_{\text{EXPORT}} = 0$	$N_{01} = 141$	$N_{00} = 774{,}106$

Plug these values into formula:

$$
\begin{aligned}
I(U;C) &= \frac{49}{801{,}948}\log_2\frac{801{,}948\cdot 49}{(49+27{,}652)(49+141)} \\
&+ \frac{141}{801{,}948}\log_2\frac{801{,}948\cdot 141}{(141+774{,}106)(49+141)} \\
&+ \frac{27{,}652}{801{,}948}\log_2\frac{801{,}948\cdot 27{,}652}{(49+27{,}652)(27{,}652+774{,}106)} \\
&+ \frac{774{,}106}{801{,}948}\log_2\frac{801{,}948\cdot 774{,}106}{(141+774{,}106)(27{,}652+774{,}106)} \\
&\approx 0.000105
\end{aligned}
$$

Linear classifiers

- Linear classifiers compute a linear combination or weighted sum $\sum_i w_i x_i$ of the feature values.
- Classification decision: $\sum_i w_i x_i > \theta$?
- Geometrically, the equation $\sum_i w_i x_i = \theta$ defines a line (2D), a plane (3D) or a hyperplane (higher dimensionalities).
- Assumption: The classes are linearly separable.
- Methods for finding a linear separator: Perceptron, Rocchio, Naive Bayes, many others

A linear classifier in 1D

- A linear separator in 1D is a point described by the equation $w_1 d_1 = \theta$

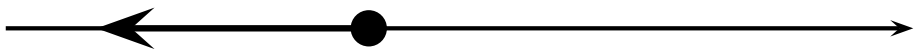
A linear classifier in 1D

- A linear separator in 1D is a point described by the equation $w_1 d_1 = \theta$
- The point at θ / w_1

A linear classifier in 1D

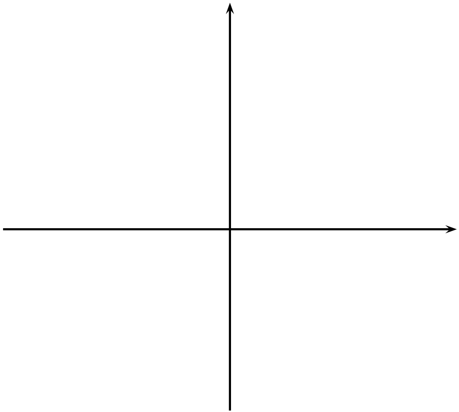
- A linear separator in 1D is a point described by the equation $w_1 d_1 = \theta$
- The point at θ / w_1
- Points (d_1) with $w_1 d_1 \geq \theta$ are in the class c.

A linear classifier in 1D



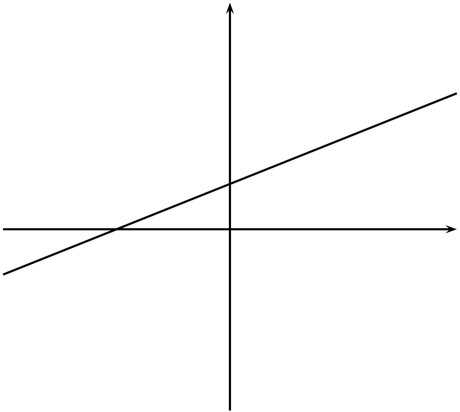
- A linear separator in 1D is a point described by the equation $w_1 d_1 = \theta$
- The point at θ / w_1
- Points (d_1) with $w_1 d_1 \geq \theta$ are in the class c.
- Points (d_1) with $w_1 d_1 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 2D



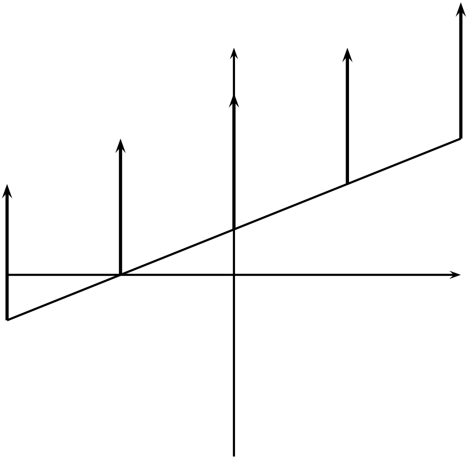
- A linear separator in 2D is a line described by the equation $w_1 d_1 + w_2 d_2 = \theta$

A linear classifier in 2D



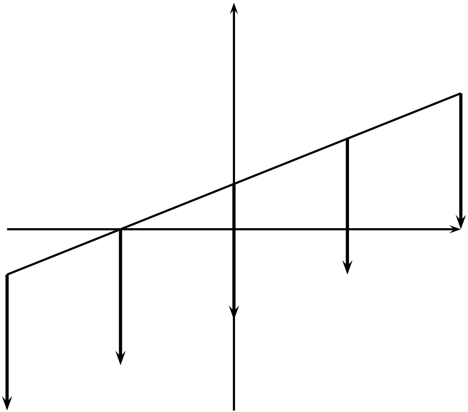
- A linear separator in 2D is a line described by the equation $w_1 d_1 + w_2 d_2 = \theta$
- Example for a 2D linear separator

A linear classifier in 2D



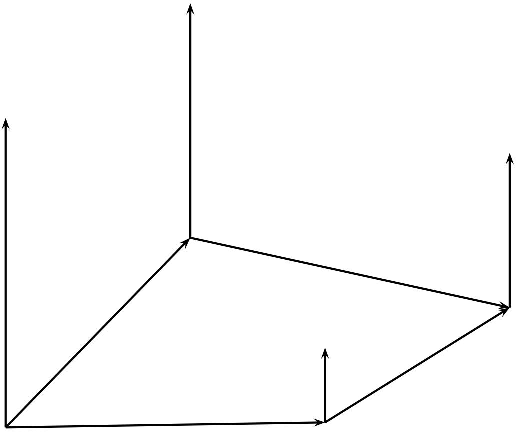
- A linear separator in 2D is a line described by the equation $w_1 d_1 + w_2 d_2 = \theta$
- Example for a 2D linear separator
- Points $(d_1 \ d_2)$ with $w_1 d_1 + w_2 d_2 \geq \theta$ are in the class c.

A linear classifier in 2D



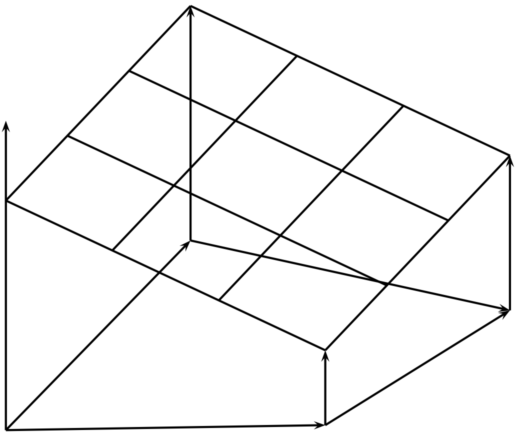
- A linear separator in 2D is a line described by the equation $w_1 d_1 + w_2 d_2 = \theta$
- Example for a 2D linear separator
- Points $(d_1 \ d_2)$ with $w_1 d_1 + w_2 d_2 \geq \theta$ are in the class c.
- Points $(d_1 \ d_2)$ with $w_1 d_1 + w_2 d_2 < \theta$ are in the complement class $\overline{c}$.

A linear classifier in 3D



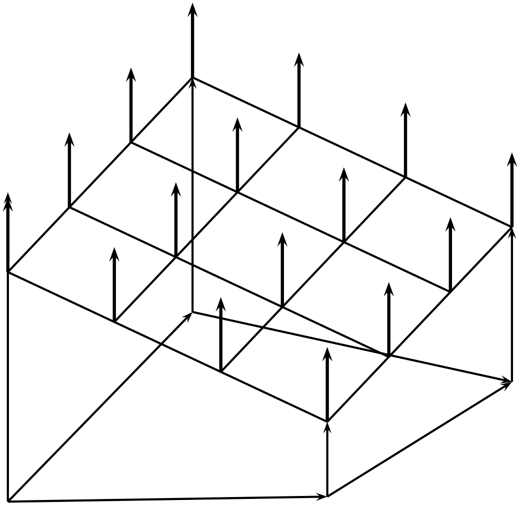
- A linear separator in 3D is a line described by the equation $w_1 d_1 + w_2 d_2 + w_3 d_3 = \theta$

A linear classifier in 3D



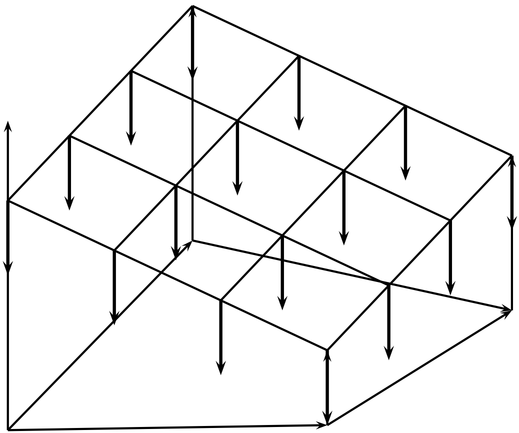
- A linear separator in 3D is a line described by the equation $w_1 d_1 + w_2 d_2 + w_3 d_3 = \theta$
- Example for a 3D linear separator

A linear classifier in 3D



- A linear separator in 3D is a line described by the equation $w_1 d_1 + w_2 d_2 + w_3 d_3 = \theta$
- Example for a 3D linear separator
- Points $(d_1\ d_2\ d_3)$ with $w_1 d_1 + w_2 d_2 + w_3 d_3 \geq \theta$ are in the class c.

A linear classifier in 3D



- A linear separator in 3D is a line described by the equation $w_1 d_1 + w_2 d_2 + w_3 d_3 = \theta$
- Example for a 3D linear separator
- Points $(d_1\ d_2\ d_3)$ with $w_1 d_1 + w_2 d_2 + w_3 d_3 \geq \theta$ are in the class c.
- Points $(d_1\ d_2\ d_3)$ with $w_1 d_1 + w_2 d_2 + w_3 d_3 < \theta$ are in the complement class $\overline{c}$.

Rocchio as a linear classifier

- Rocchio is a linear separator defined by:

$$\sum_{i=1}^{M} w_i d_i = \vec{w}\vec{d} = \theta$$

where the normal vector $\vec{w} = \vec{\mu}(c_1) - \vec{\mu}(c_2)$ and $\theta = 0.5 * (|\vec{\mu}(c_1)|^2 - |\vec{\mu}(c_2)|^2)$.

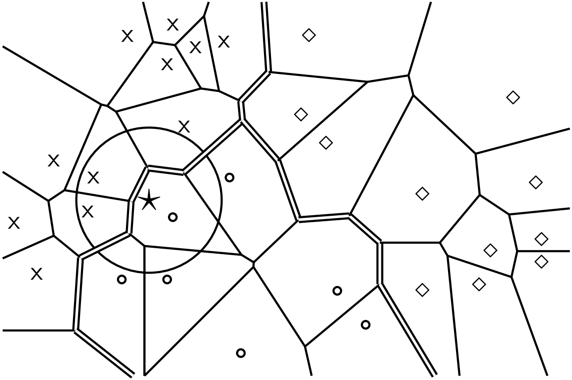
Naive Bayes as a linear classifier

Naive Bayes is a linear separator defined by:

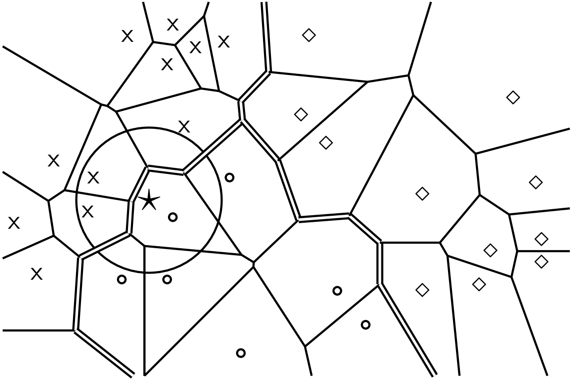
$$\sum_{i=1}^{M} w_i d_i = \theta$$

where $w_i = \log[\hat{P}(t_i|c)/\hat{P}(t_i|\bar{c})]$, d_i = number of occurrences of t_i in d, and $\theta = -\log[\hat{P}(c)/\hat{P}(\bar{c})]$. Here, the index i, $1 \leq i \leq M$, refers to terms of the vocabulary (not to positions in d as k did in our original definition of Naive Bayes)

kNN is not a linear classifier

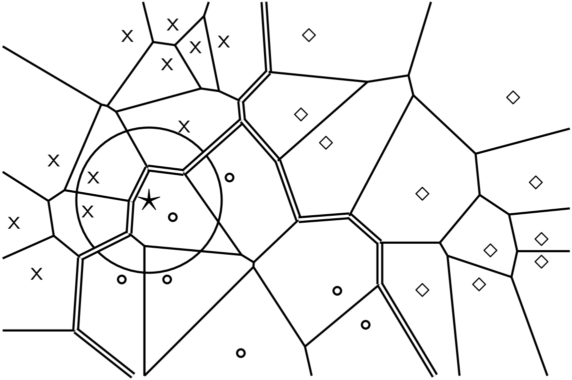


kNN is not a linear classifier



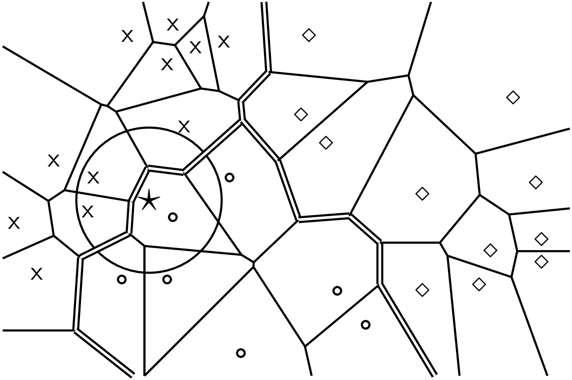
- Classification decision based on majority of k nearest neighbors.

kNN is not a linear classifier



- Classification decision based on majority of k nearest neighbors.
- The decision boundaries between classes are piecewise linear . . .

kNN is not a linear classifier



- Classification decision based on majority of k nearest neighbors.
- The decision boundaries between classes are piecewise linear ...
- ... but they are not linear separators that can be described as $\sum_{i=1}^{M} w_i d_i = \theta$.

Outline

What is clustering?

- Clustering is the process of grouping a set of documents into clusters of similar documents.

What is clustering?

- Clustering is the process of grouping a set of documents into clusters of similar documents.
- Documents within a cluster should be similar.

What is clustering?

- 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.

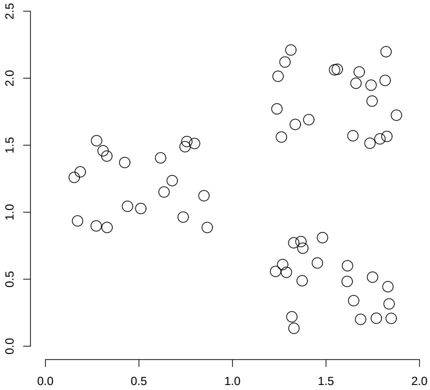
What is clustering?

- 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.

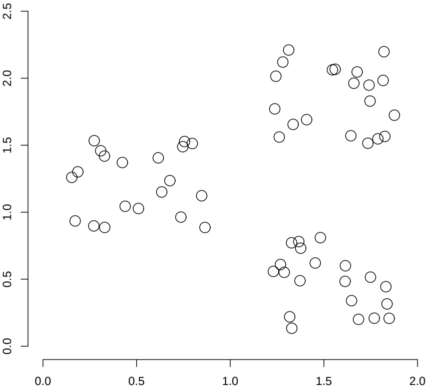
What is clustering?

- 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



Data set with clear cluster structure



How would you design an algorithm for finding the three clusters in this case?

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

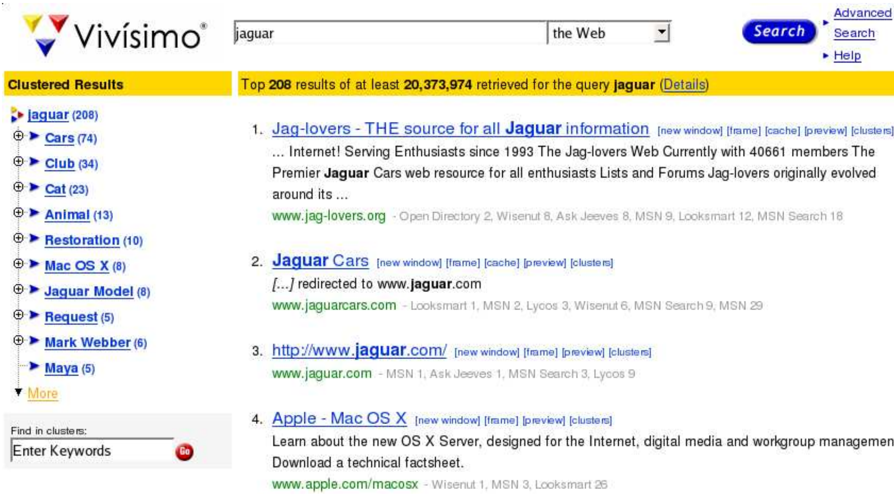
Cluster hypothesis. Documents in the same cluster behave similarly with respect to relevance to information needs.

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

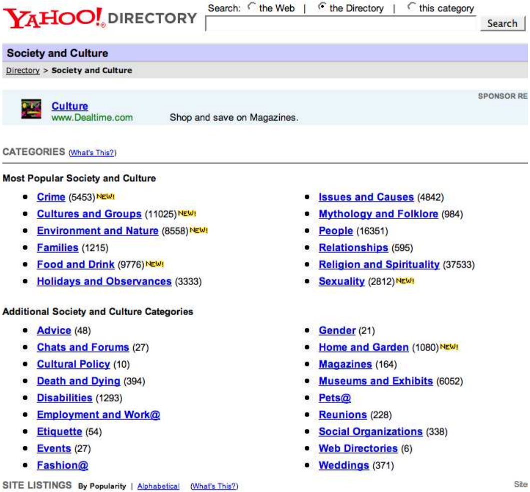
Applications of clustering in IR

Application	What is clustered?	Benefit	Example
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	McKeown et al. 2002, http://news.google.com
Language modeling	collection	increased precision and/or recall	Liu&Croft 2004
Cluster-based retrieval	collection	higher efficiency: faster search	Salton 1971

Search result clustering for better navigation



Global navigation: Yahoo



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)

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

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

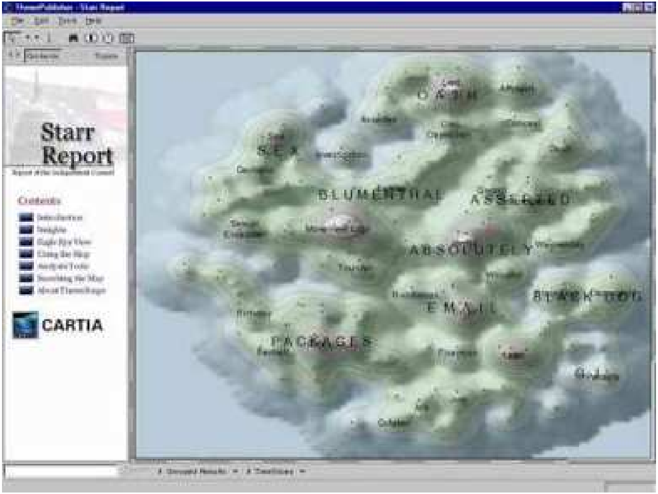
- Note: Yahoo/MESH are not examples of clustering.
- But they are well known examples for using a global hierarchy for navigation.
- Global navigation based on clustering:

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

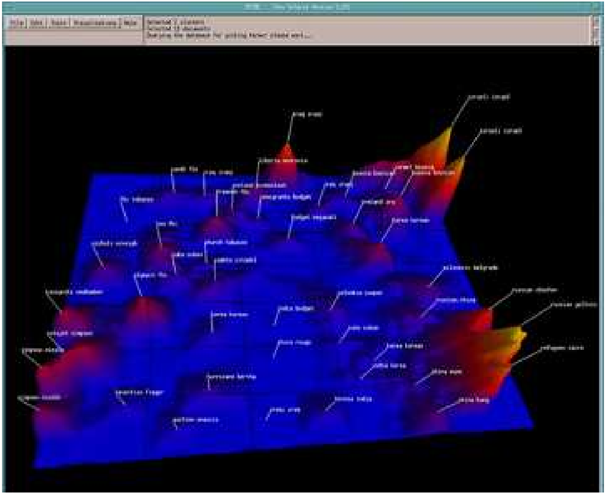
- Note: Yahoo/MESH are not examples of clustering.
- But they are well known examples for using a global hierarchy for navigation.
- Global navigation based on clustering:
  - Cartia
  - Themescapes

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

Global navigation combined with visualization (1)



Global navigation combined with visualization (2)



Global clustering for navigation: Google News

http://news.google.com

Clustering for improving recall

- To improve search recall:

Clustering for improving recall

- To improve search recall:
  - Cluster docs in collection a priori

Clustering for improving recall

- To improve search recall:
  - Cluster docs in collection a priori
  - When a query matches a doc d, also return other docs in the cluster containing d

Clustering for improving recall

- To improve search recall:
  - Cluster docs in collection a priori
  - When a query matches a doc d, also return other docs in the cluster containing d
- Hope if we do this: the query "car" will also return docs containing "automobile"

Clustering for improving recall

- To improve search recall:
  - Cluster docs in collection a priori
  - When a query matches a doc d, also return other docs in the cluster containing d
- Hope if we do this: the query "car" will also return docs containing "automobile"
  - Because clustering grouped together docs containing "car" with those containing "automobile".

Clustering for improving recall

- To improve search recall:
  - Cluster docs in collection a priori
  - When a query matches a doc d, also return other docs in the cluster containing d
- Hope if we do this: the query "car" will also return docs containing "automobile"
  - Because clustering grouped together docs containing "car" with those containing "automobile".
  - Why?

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 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 equivalent to cosine similarity.
- Recall: centroids are not length-normalized.

Document representations in clustering

- Vector space model
- As in vector space classification, we measure relatedness between vectors by Euclidean distance . . .
- . . . which is equivalent to cosine similarity.
- Recall: centroids are not length-normalized.
- For centroids, distance and cosine give different results.

Issues in clustering

- How many clusters?

Issues in clustering

- How many clusters?
- Initially, we will assume the number of clusters K is given.

Issues in clustering

- How many clusters?
- Initially, we will assume the number of clusters K is given.
- General goal: put related docs in the same cluster, put unrelated docs in different clusters.

Issues in clustering

- How many clusters?
- Initially, we will assume the number of clusters K is given.
- General goal: put related docs in the same cluster, put unrelated docs in different clusters.
- But how do we formalize this?

Issues in clustering

- How many clusters?
- Initially, we will assume the number of clusters K is given.
- General goal: put related docs in the same cluster, put unrelated docs in different clusters.
- But how do we formalize this?
- Often: secondary goals in clustering

Issues in clustering

- How many clusters?
- Initially, we will assume the number of clusters K is given.
- General goal: put related docs in the same cluster, put unrelated docs in different clusters.
- But how do we formalize this?
- Often: secondary goals in clustering
    - Example: avoid very small and very large clusters

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 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 a pair of sneakers in two clusters: (i) sports apparel and (ii) 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 a pair of sneakers in two clusters: (i) sports apparel and (ii) 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 a pair of sneakers in two clusters: (i) sports apparel and (ii) shoes
  - You can only do that with a soft clustering approach.
- We won't have time for soft clustering. See IIR 16.5, IIR 18

Our plan

- This lecture: Flat, hard clustering

Our plan

- This lecture: Flat, hard clustering
- Next lecture: Hierarchical, hard clustering

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 in 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 in 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 in 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 in 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

- Objective/partitioning criterion: minimize the average squared difference from the centroid

K-means

- 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

- 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

- 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

- 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 algorithm

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

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

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

K-means example

Convergence of K-means

- K-means converges to a fixed point in a finite number of iterations.

Convergence of K-means

- K-means converges to a fixed point in a finite number of iterations.
- Proof:

Convergence of K-means

- K-means converges to a fixed point in a finite number of iterations.
- Proof:
  - The sum of squared distances (RSS) decreases during reassignment.

Convergence of *K*-means

- *K*-means converges to a fixed point in a finite number of iterations.
- Proof:
  - The sum of squared distances (RSS) decreases during reassignment.
  - (because each vector is moved to a closer centroid)

Convergence of K-means

- K-means converges to a fixed point in a finite number of iterations.
- Proof:
  - The sum of squared distances (RSS) decreases during reassignment.
  - (because each vector is moved to a closer centroid)
  - RSS decreases during recomputation.

Convergence of K-means

- K-means converges to a fixed point in a finite number of iterations.
- Proof:
  - The sum of squared distances (RSS) decreases during reassignment.
  - (because each vector is moved to a closer centroid)
  - RSS decreases during recomputation.
  - (We will show this on the next slide.)

Convergence of K-means

- K-means converges to a fixed point in a finite number of iterations.
- Proof:
  - The sum of squared distances (RSS) decreases during reassignment.
  - (because each vector is moved to a closer centroid)
  - RSS decreases during recomputation.
  - (We will show this on the next slide.)
  - There is only a finite number of clusterings.

Convergence of K-means

- K-means converges to a fixed point in a finite number of iterations.
- Proof:
  - The sum of squared distances (RSS) decreases during reassignment.
  - (because each vector is moved to a closer centroid)
  - RSS decreases during recomputation.
  - (We will show this on the next slide.)
  - There is only a finite number of clusterings.
  - Thus: We must reach a fixed point.

Convergence of *K*-means

- *K*-means converges to a fixed point in a finite number of iterations.
- Proof:
  - The sum of squared distances (RSS) decreases during reassignment.
  - (because each vector is moved to a closer centroid)
  - RSS decreases during recomputation.
  - (We will show this on the next slide.)
  - There is only a finite number of clusterings.
  - Thus: We must reach a fixed point.
  - (assume that ties are broken consistently)

Convergence of K-means

- K-means converges to a fixed point in a finite number of iterations.
- Proof:
  - The sum of squared distances (RSS) decreases during reassignment.
  - (because each vector is moved to a closer centroid)
  - RSS decreases during recomputation.
  - (We will show this on the next slide.)
  - There is only a finite number of clusterings.
  - Thus: We must reach a fixed point.
  - (assume that ties are broken consistently)
- But we don't know how long convergence will take!

Convergence of *K*-means

- *K*-means converges to a fixed point in a finite number of iterations.
- Proof:
  - The sum of squared distances (RSS) decreases during reassignment.
  - (because each vector is moved to a closer centroid)
  - RSS decreases during recomputation.
  - (We will show this on the next slide.)
  - There is only a finite number of clusterings.
  - Thus: We must reach a fixed point.
  - (assume that ties are broken consistently)
- 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).

Convergence of K-means

- K-means converges to a fixed point in a finite number of iterations.
- Proof:
  - The sum of squared distances (RSS) decreases during reassignment.
  - (because each vector is moved to a closer centroid)
  - RSS decreases during recomputation.
  - (We will show this on the next slide.)
  - There is only a finite number of clusterings.
  - Thus: We must reach a fixed point.
  - (assume that ties are broken consistently)
- 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).
- But complete convergence can take many more iterations.

Recomputation decreases average distance

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

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

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

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!

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.

Optimality of K-means

- Convergence does not mean that we converge to the optimal clustering!

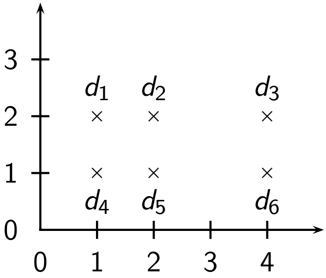
Optimality of K-means

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

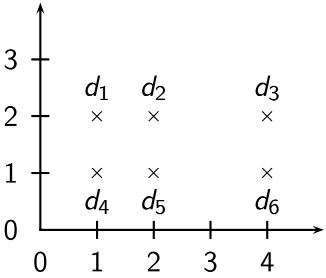
Optimality of K-means

- 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.

Example for suboptimal clustering

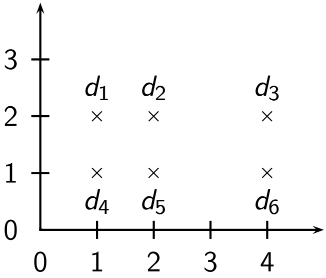


Example for suboptimal clustering



- What is the optimal clustering for $K = 2$?

Example for suboptimal clustering



- What is the optimal clustering for $K = 2$?
- Do we converge on this clustering for arbitrary seeds d_{i_1}, d_{i_2}?

Initialization of K-means

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

Initialization of K-means

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

Initialization of K-means

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

Initialization of K-means

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

- Seed selection is just one of many ways K-means can be initialized.
- Seed selection is not very robust: It's easy to get a suboptimal clustering.
- Better heuristics:
  - 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 (next class)

Initialization of K-means

- Seed selection is just one of many ways K-means can be initialized.
- Seed selection is not very robust: It's easy to get a suboptimal clustering.
- Better heuristics:
  - 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 (next class)
  - Select i (e.g., $i = 10$) different sets of seeds, do a K-means clustering for each, select the clustering with lowest RSS

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 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 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 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 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 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, the number of iterations can be much higher than linear in the number of documents.

Outline

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

- 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

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

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

External criterion: Purity

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

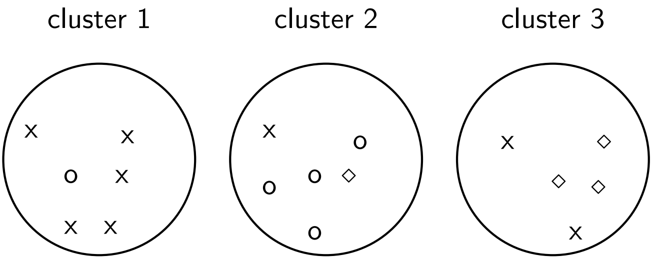
- $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ is the set of clusters and $C = \{c_1, c_2, \ldots, c_J\}$ is the set of classes.
- For each cluster ω_k: find class c_j with most members n_{kj} in ω_k

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



Majority class and number of members of the majority class for the three clusters are: x, 5 (cluster 1); o, 4 (cluster 2); and $\diamond$, 3 (cluster 3). Purity is $(1/17) \times (5 + 4 + 3) \approx 0.71$.

Rand index

- Definition: $\mathrm{RI} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}$

Rand index

- Definition: $\text{RI} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}}$
- Based on 2x2 contingency table:

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

Rand index

- Definition: $\text{RI} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}}$
- Based on 2x2 contingency table:

	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.

Rand index

- Definition: $\mathrm{RI} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}$
- Based on 2x2 contingency table:

	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.
- There are $\binom{N}{2}$ pairs for N documents.

Rand index

- Definition: $\mathrm{RI} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}$
- Based on 2x2 contingency table:

	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.
- There are $\binom{N}{2}$ pairs for N documents.
- Example: $\binom{13}{2} = 136$ in o/$\diamond$/x example

Rand index

- Definition: $\mathrm{RI} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}$
- Based on 2x2 contingency table:

	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.
- There are $\binom{N}{2}$ pairs for N documents.
- Example: $\binom{13}{2} = 136$ 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) . . .

Rand index

- Definition: $\text{RI} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}}$
- Based on 2x2 contingency table:

	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.
- There are $\binom{N}{2}$ pairs for N documents.
- Example: $\binom{13}{2} = 136$ 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.

As an example, we compute RI for the o/$\diamond$/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 $\diamond$ 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/◇/x example

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

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

Evaluation results for the o/$\diamond$/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).

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

Outline

How many clusters?

- Either: Number of clusters K is given.

How many clusters?

- Either: Number of clusters K is given.
    - Then partition into K clusters

How many clusters?

- Either: Number of clusters K is given.
  - Then partition into K clusters
  - K might be given because there is some external constraint. 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?

- Either: Number of clusters K is given.
  - Then partition into K clusters
  - K might be given because there is some external constraint. Example: In the case of Scatter-Gather, it was hard to show more than 10–20 clusters on a monitor in the 90s.
- Or: Finding the "right" number of clusters is part of the problem.

How many clusters?

- Either: Number of clusters K is given.
  - Then partition into K clusters
  - K might be given because there is some external constraint. Example: In the case of Scatter-Gather, it was hard to show more than 10–20 clusters on a monitor in the 90s.
- Or: Finding the "right" number of clusters is part of the problem.
  - Given docs, find K for which an optimum is reached.

How many clusters?

- Either: Number of clusters K is given.
  - Then partition into K clusters
  - K might be given because there is some external constraint. Example: In the case of Scatter-Gather, it was hard to show more than 10–20 clusters on a monitor in the 90s.
- Or: Finding the "right" number of clusters is part of the problem.
  - Given docs, find K for which an optimum is reached.
  - How to define "optimum"?

How many clusters?

- Either: Number of clusters K is given.
  - Then partition into K clusters
  - K might be given because there is some external constraint. Example: In the case of Scatter-Gather, it was hard to show more than 10–20 clusters on a monitor in the 90s.
- Or: Finding the "right" number of clusters is part of the problem.
  - Given docs, find K for which an optimum is reached.
  - How to define "optimum"?
  - Why can't we use RSS or average squared distance from centroid?

Simple objective function for K (1)

- Basic idea:

Simple objective function for K (1)

- Basic idea:
  - Start with 1 cluster ($K = 1$)

Simple objective function for K (1)

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

Simple objective function for K (1)

- 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 (1)

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

Simple objective function for K (1)

- Basic idea:
  - Start with 1 cluster ($K = 1$)
  - Keep adding clusters (= keep increasing K)
  - Add a penalty for each new cluster
- Trade off cluster penalties against average squared distance from centroid
- Choose K with best tradeoff

Simple objective function for K (2)

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

Simple objective function for K (2)

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

Simple objective function for K (2)

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

Simple objective function for K (2)

- Given a clustering, define the cost for a document as (squared) distance to centroid
- Define total distortion RSS(K) as sum of all invididual 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 (2)

- Given a clustering, define the cost for a document as (squared) distance to centroid
- Define total distortion RSS(K) as sum of all invididual 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) $+ K\lambda$

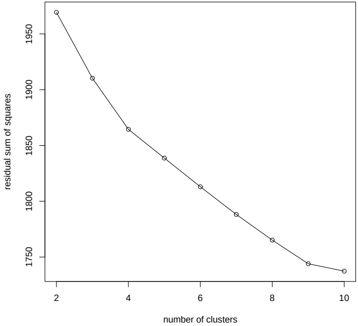
Simple objective function for K (2)

- Given a clustering, define the cost for a document as (squared) distance to centroid
- Define total distortion RSS(K) as sum of all invididual 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 (2)

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



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

Resources

- Chapter 16 of IIR

Resources

- Chapter 16 of IIR
- Resources at `http://ifnlp.org/ir`

Resources

- Chapter 16 of IIR
- Resources at `http://ifnlp.org/ir`
- K-means example

Resources

- Chapter 16 of IIR
- Resources at `http://ifnlp.org/ir`
- K-means example
- Keith van Rijsbergen on the cluster hypothesis (he was one of the originators)

Resources

- Chapter 16 of IIR
- Resources at `http://ifnlp.org/ir`
- K-means example
- Keith van Rijsbergen on the cluster hypothesis (he was one of the originators)
- Clusty/Vivisimo: search result clustering