Overview

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

MI example for poultry/export in Reuters

\[ I(U; C) = \frac{49}{801,948} \log_2 \frac{801,948 \cdot 49}{(49+27,652)(49+141)} + \frac{141}{801,948} \log_2 \frac{801,948 \cdot 141}{(141+774,106)(49+141)} + \frac{27,652}{801,948} \log_2 \frac{801,948 \cdot 27,652}{(27,652+774,106)(49+27,652)} + \frac{774,106}{801,948} \log_2 \frac{801,948 \cdot 774,106}{(27,652+774,106)(27,652+774,106)} \approx 0.000105 \]
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 $w_1 d_1 = \theta$

The point at $\theta/w_1$

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 $\bar{c}$.

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 $\bar{c}$.

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 $\bar{c}$. 
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 \( \theta = 0.5 \times (|\vec{\mu}(c_1)|^2 - |\vec{\mu}(c_2)|^2) \).

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

Classification decision based on majority of \( k \) nearest neighbors.
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 $\sum_{i=1}^{M} w_i d_i = \theta$. 

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

**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, ...
## Applications of clustering in IR

<table>
<thead>
<tr>
<th>Application</th>
<th>What is clustered?</th>
<th>Benefit</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search result clustering</td>
<td>search results</td>
<td>more effective information presentation to user</td>
<td>McKeown et al. 2002, <a href="http://news.google.com">http://news.google.com</a></td>
</tr>
<tr>
<td>Scatter-Gather</td>
<td>(subsets of)</td>
<td>alternative user interface: “search without typing”</td>
<td>Liu &amp; Croft 2004</td>
</tr>
<tr>
<td>Collection clustering</td>
<td>collection</td>
<td>effective information presentation for exploratory browsing</td>
<td>Salton 1971</td>
</tr>
<tr>
<td>Language modeling</td>
<td>collection</td>
<td>increased precision and/or recall</td>
<td>Salton 1971</td>
</tr>
<tr>
<td>Cluster-based retrieval</td>
<td>collection</td>
<td>higher efficiency: faster search</td>
<td>Salton 1971</td>
</tr>
</tbody>
</table>

---

## Search result clustering for better navigation

1. **Jag-lovers** - THE source for all Jaguar information

   - internet search: Efficient since 1999 The Jag-lovers Web currently has 40961 members. The premier Jaguar Cars web resource for all enthusiasts is the Jaguar-lovers.com site. 

2. **Jaguar Cars**

   - [www.jaguar.com](http://www.jaguar.com): The new Jaguar website is designed for the Internet, digital media and real-world management. Read a technical review.
   - www.jaguar.com

3. **Apple** - Mac OS X

   - Learn about the new OS X Server, designed for the Internet, digital media and real-world management. Read a technical review.
   - www.apple.com/macosx

---

## Global navigation: Yahoo

- **Society and Culture**
- **Culture**
- **Additional Society and Culture Categories**
  - Advice
  - Charity
  - Children and Family
  - Crime
  - Cultures and Usages
  - Environment and Nature
  - Families
  - Food and Drink
  - Holidays and Observances

## Global navigation: MESH (upper level)

- **MeSH 2008**

---

Inset image: Yahoo and MESH tree structures.
Global navigation: MESH (lower level)

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:
  - 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?

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

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.
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.
  - 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
- Next lecture: Hierarchical, hard clustering

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

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

K-means

- Objective/partitioning criterion: minimize the average squared difference from the centroid
- Recall definition of centroid:
  \[ \bar{\mu}(\omega) = \frac{1}{|\omega|} \sum_{\bar{x} \in \omega} \bar{x} \]
  
  where we use \( \omega \) 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: recomputate each centroid as the average of the vectors that were assigned to it in reassignment

K-means algorithm

\[
\text{K-MEANS}((\bar{x}_1, \ldots, \bar{x}_N), K) \\
1 \quad (\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_K) \leftarrow \text{SELECTRANDOMSEEDS}((\bar{x}_1, \ldots, \bar{x}_N), K) \\
2 \quad \text{for } k \leftarrow 1 \text{ to } K \\
3 \quad \quad \mu_k \leftarrow \bar{s}_k \\
4 \quad \text{while stopping criterion has not been met} \\
5 \quad \text{for } k \leftarrow 1 \text{ to } K \\
6 \quad \quad \omega_k \leftarrow \{\} \\
7 \quad \quad \text{for } n \leftarrow 1 \text{ to } N \\
8 \quad \quad \quad j \leftarrow \arg \min_{j'} |\mu_{j'} - \bar{x}_n| \\
9 \quad \quad \quad \omega_j \leftarrow \omega_j \cup \{\bar{x}_n\} \quad \text{(reassignment of vectors)} \\
10 \quad \quad \text{for } k \leftarrow 1 \text{ to } K \\
11 \quad \quad \mu_k \leftarrow \frac{1}{|\omega_k|} \sum_{\bar{x} \in \omega_k} \bar{x} \quad \text{(recomputation of centroids)} \\
12 \quad \text{return } \{\mu_1, \ldots, \mu_K\}
\]
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.

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.

Recomputation decreases average distance

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

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

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

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

The last line is the componentwise definition of the centroid! 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.

Example for suboptimal clustering

- What is the optimal clustering for $K = 2$?
- Do we converge on this clustering for arbitrary seeds $d_{i1}, d_{i2}$?
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 $O(M)$.
- 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: $O(INKM)$ – 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

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

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.
- 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

\[ \text{purity}(\Omega, C) = \frac{1}{N} \sum_{k} \max_{j} |\omega_k \cap c_j| \]

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

Example for computing purity

\[ \begin{array}{ccc}
\text{cluster 1} & \text{cluster 2} & \text{cluster 3} \\
\bigcirc x x & \bigcirc x \bigcirc & \bigcirc x \bigcirc \\
\bigcirc x x & \bigcirc & \\
\end{array} \]

Majority class

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

Rand index

- Definition: \( \text{RI} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}} \)
- Based on 2x2 contingency table:
  \( \begin{array}{cc}
  \text{same cluster} & \text{different clusters} \\
  \text{same class} & \text{true positives (TP)} & \text{false negatives (FN)} \\
  \text{different classes} & \text{false positives (FP)} & \text{true negatives (TN)} \\
  \end{array} \)
- \( \text{TP}+\text{FN}+\text{FP}+\text{TN} \) is the total number of pairs.
- There are \( \binom{N}{2} \) pairs for \( N \) documents.
- Example: \( \binom{13}{2} = 136 \) in \( \bigcirc/\bigcirc/x \) 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/⋄/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:

\[
TP + FP = \binom{6}{2} + \binom{6}{2} + \binom{5}{2} = 40
\]

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:

\[
TP = \binom{5}{2} + \binom{4}{2} + \binom{3}{2} + \binom{2}{2} = 20
\]

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

Rand measure for the o/⋄/x example

\[
\text{same cluster} \quad \text{different clusters}
\]

\[
\begin{array}{cc}
\text{same class} & \text{different classes} \\
TP & 20 \\
FN & 24 \\
FP & 20 \\
TN & 72 \\
\end{array}
\]

RI is then

\[
(20 + 72)/(20 + 20 + 24 + 72) \approx 0.68.
\]

Evaluation results for the o/⋄/x example

<table>
<thead>
<tr>
<th>purity</th>
<th>NMI</th>
<th>RI</th>
<th>F5</th>
</tr>
</thead>
<tbody>
<tr>
<td>lower bound</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>maximum</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>value for example</td>
<td>0.71</td>
<td>0.36</td>
<td>0.68</td>
</tr>
</tbody>
</table>

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

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
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
- Define total distortion $RSS(K)$ as sum of all individual document costs (corresponds to average distance)
- Then: penalize each cluster with a cost $\lambda$
- 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 $\lambda$...
Finding the “knee” in the curve

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

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