Data-analysis and Retrieval
Clustering

Ad Feelders

June 12, 2020

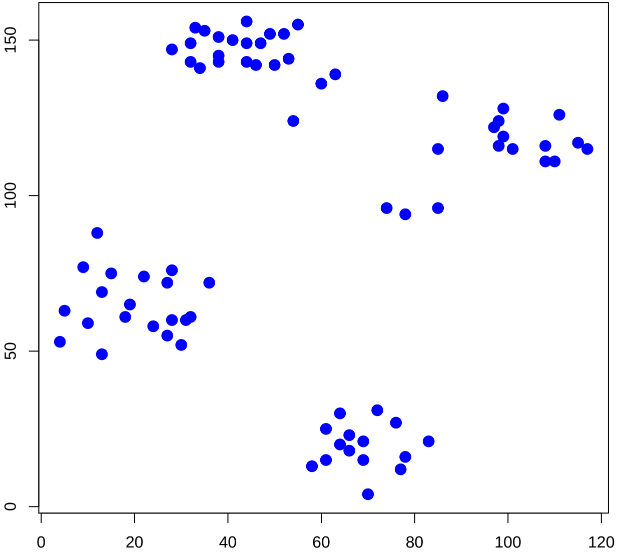
Overview

1. What is clustering?
2. Applications of clustering in information retrieval
3. K-means algorithm
4. Evaluation of clustering
5. How many clusters?

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.

How many clusters?



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, ...

The cluster hypothesis

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

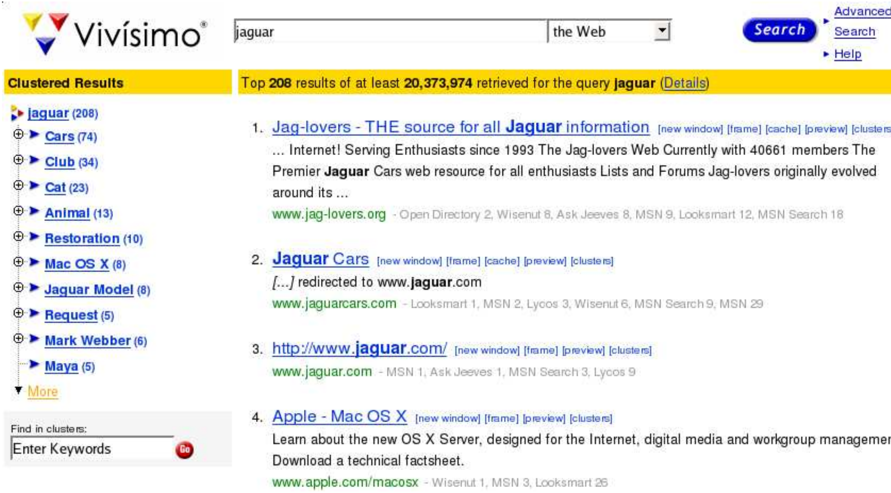
All applications of clustering in IR are based (directly or indirectly) on the cluster hypothesis.

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

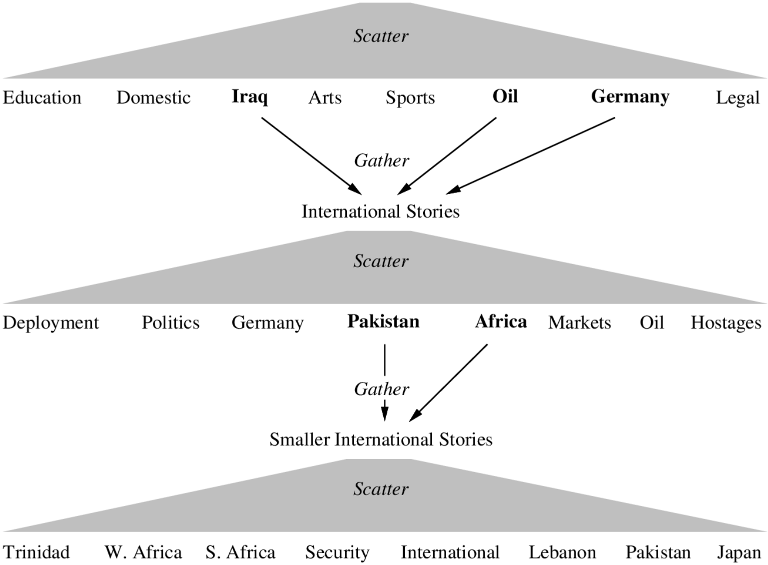
Applications of clustering in IR

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



Scatter-Gather



About Google News

Google News is a computer-generated news service that aggregates headlines from more than 50,000 news sources worldwide, *groups similar stories together*, and displays them according to each reader's interests.

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 the clustering algorithm groups together docs containing "car" with those containing "automobile".
  - Both types of documents contain words like "parts", "dealer", "mercedes", "road trip".
- Speed up search: match query to cluster centers.

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 algorithms
  - Usually start with a random partitioning of docs into groups
  - Refine iteratively
  - Main algorithm: K-means
- Hierarchical algorithms (not discussed)
  - 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 *sneakers* in two clusters:
    - sports apparel
    - shoes
  - You can only do that with a soft clustering approach.

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: $O(K^N)$ possible partitions.
- Effective heuristic method: K-means algorithm

K-means

- Perhaps the best known clustering algorithm
- Vector space model
- We measure relatedness between vectors by Euclidean distance.
- ...which is almost equivalent to cosine similarity, if document vectors are normalized to unit length.
- Almost: centroids are not length-normalized.

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:

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

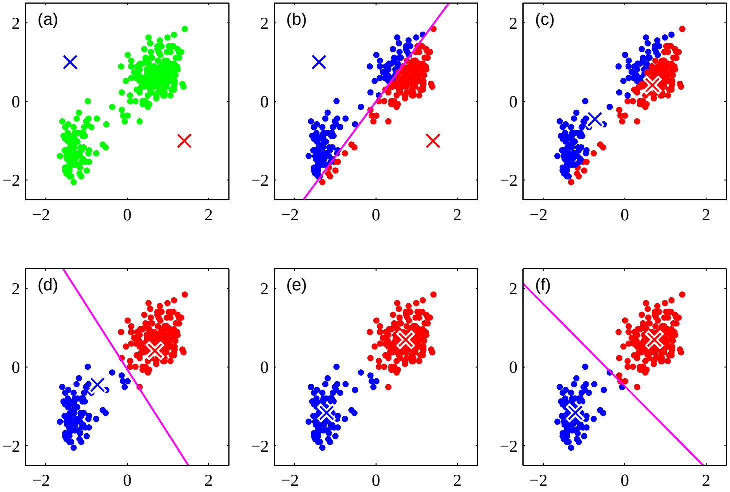
  where we use ω to denote a cluster.
- We try to find the minimum average squared difference by iterating two steps:
  - reassignment: assign each vector to its closest centroid
  - recomputation: recompute each centroid as the average of the vectors that were assigned to it

K-means algorithm

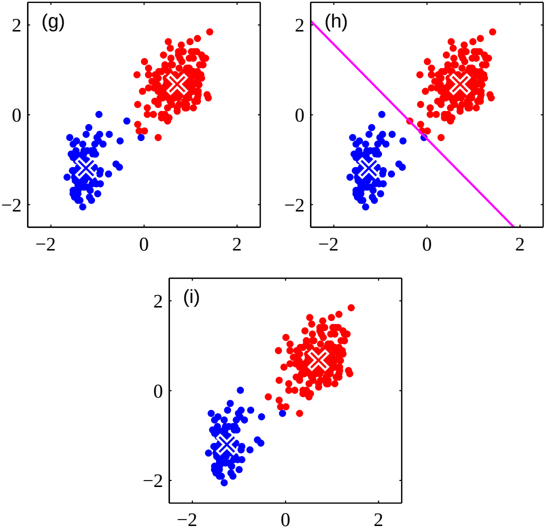
1. Partition the observations into K initial clusters.
2. Calculate the mean of each cluster.
3. Assign each observation to the cluster whose mean is nearest.
4. If reassignments have taken place, return to step 2; otherwise stop.

Old Faithful data set

Old Faithful data set: $K = 2$



Old Faithful data set $K = 2$



K-means is guaranteed to converge

- RSS = sum of all squared distances between document vector and assigned centroid
- RSS decreases during each reassignment step.
  - because each vector is moved to its closest 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 $\rightarrow$ convergence

Recomputation decreases average distance

The objective function to minimize is:

$$\mathrm{RSS} = \sum_{k=1}^{K} \mathrm{RSS}_k$$

Objective function for cluster k:

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

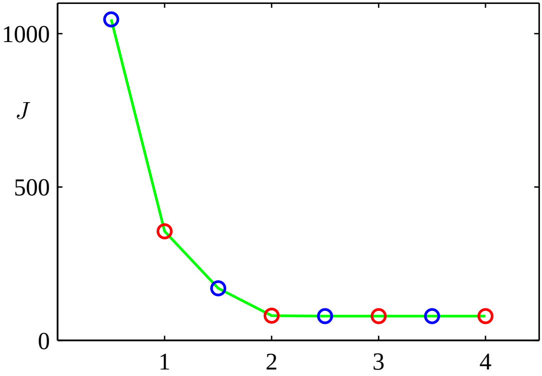
Take derivative with respect to v_j and equate to zero:

$$\begin{aligned}
\frac{\partial \mathrm{RSS}_k(\vec{v})}{\partial v_j} &= \sum_{\vec{x} \in \omega_k} 2(v_j - x_j) = 0 \\
|\omega_k| \; v_j &= \sum_{\vec{x} \in \omega_k} x_j \\
v_j &= \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} x_j
\end{aligned}$$

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

Convergence of algorithm on Old Faithful

RSS on vertical axis; iteration on horizontal axis



K-means is guaranteed to converge

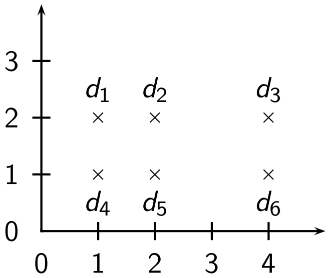
- But we don't know how long convergence will take!
- If we don't care about a few documents switching back and forth, then convergence is usually fast ($<$ 10-20 iterations).
- However, complete convergence can take many more iterations.

Optimality of K-means

- Convergence $\neq$ optimality
- 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

	x_1	x_2
d_1	1	2
d_2	2	2
d_3	4	2
d_4	1	1
d_5	2	1
d_6	4	1



- What happens if we use d_2 and d_5 as initial cluster centers?
- What if we use d_2 and d_3?

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 for example 10 different random sets of seeds, do a K-means clustering for each, select the clustering with lowest RSS

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

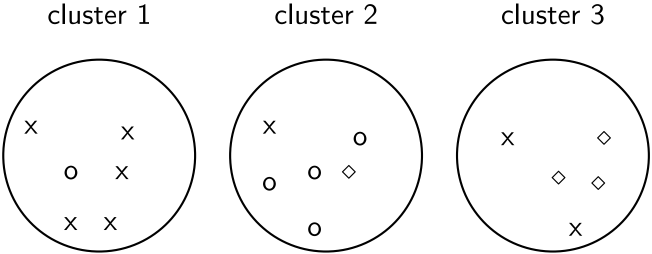
- Based on a gold standard data set, e.g., the Reuters collection we also used for the evaluation of classification
- Goal: Clustering should reproduce the classes in the gold standard
- (But we only want to reproduce how documents are divided into groups, not the class labels.)
- First measure for how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

External criterion: Purity

$$\text{purity}(\Omega, C) = \frac{1}{N} \sum_{k=1}^{K} \max_{j \in \{1,\ldots,J\}} |\omega_k \cap c_j|$$

- $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ 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 of purity computation



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 $(1/17) \times (5 + 4 + 3) \approx 0.71$.

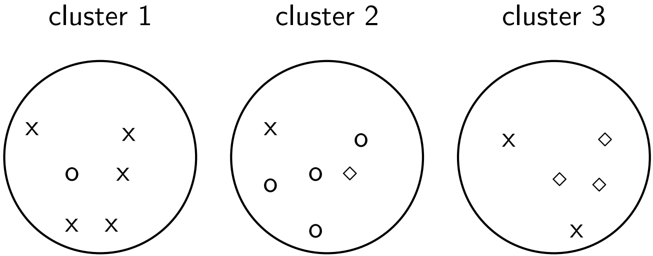
Another external criterion: Rand index

- 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.

Example of rand index computation



Cluster 1 contains $\binom{6}{2} = \frac{6 \times 5}{2} = 15$ positives.

Cluster 3 contains $\binom{3}{2} + \binom{2}{2} = 3 + 1 = 4$ true positives.

How many true positives are there in cluster 1?

Rand Index: Example

As an example, we compute RI for the o/$\diamond$/x example. We first compute $\mathrm{TP} + \mathrm{FP}$. 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:

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

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:

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

Thus, $\mathrm{FP} = 40 - 20 = 20$.
FN and TN are computed similarly.

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

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.

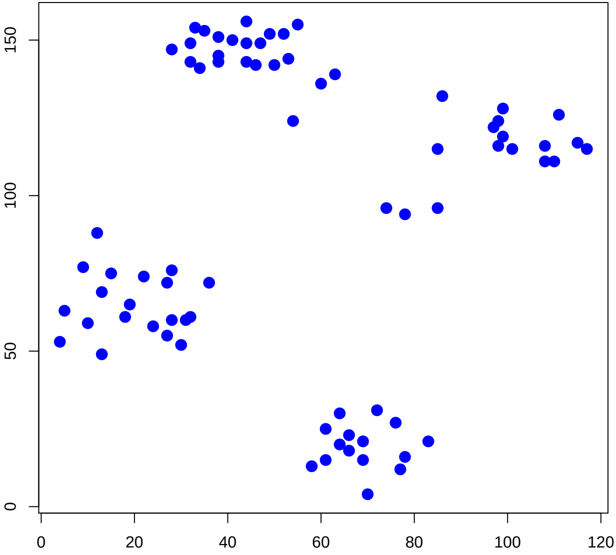
Simple objective function for K: Basic idea

- 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

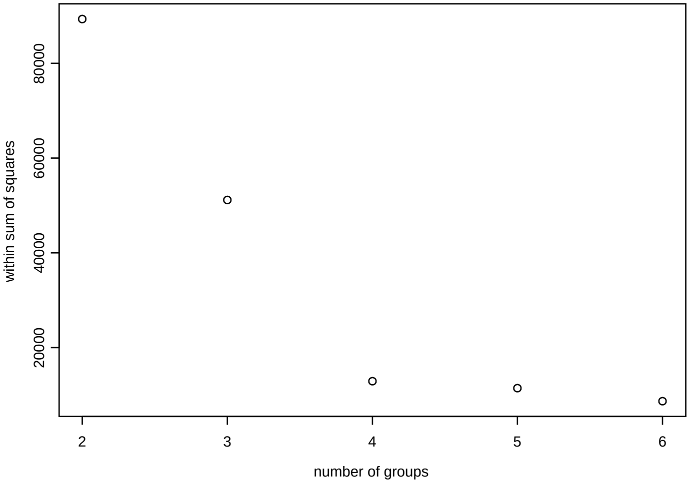
Simple objective function for K: Formalization

- 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: $\mathrm{RSS(K)} + K\lambda$
- Select K that minimizes $(\mathrm{RSS(K)} + K\lambda)$
- Still need to determine good value for λ ...

Data set with clear cluster structure



Rule of thumb: find the "hinge point" in the curve



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