
BLG 540E TEXT RETRIEVAL SYSTEMS

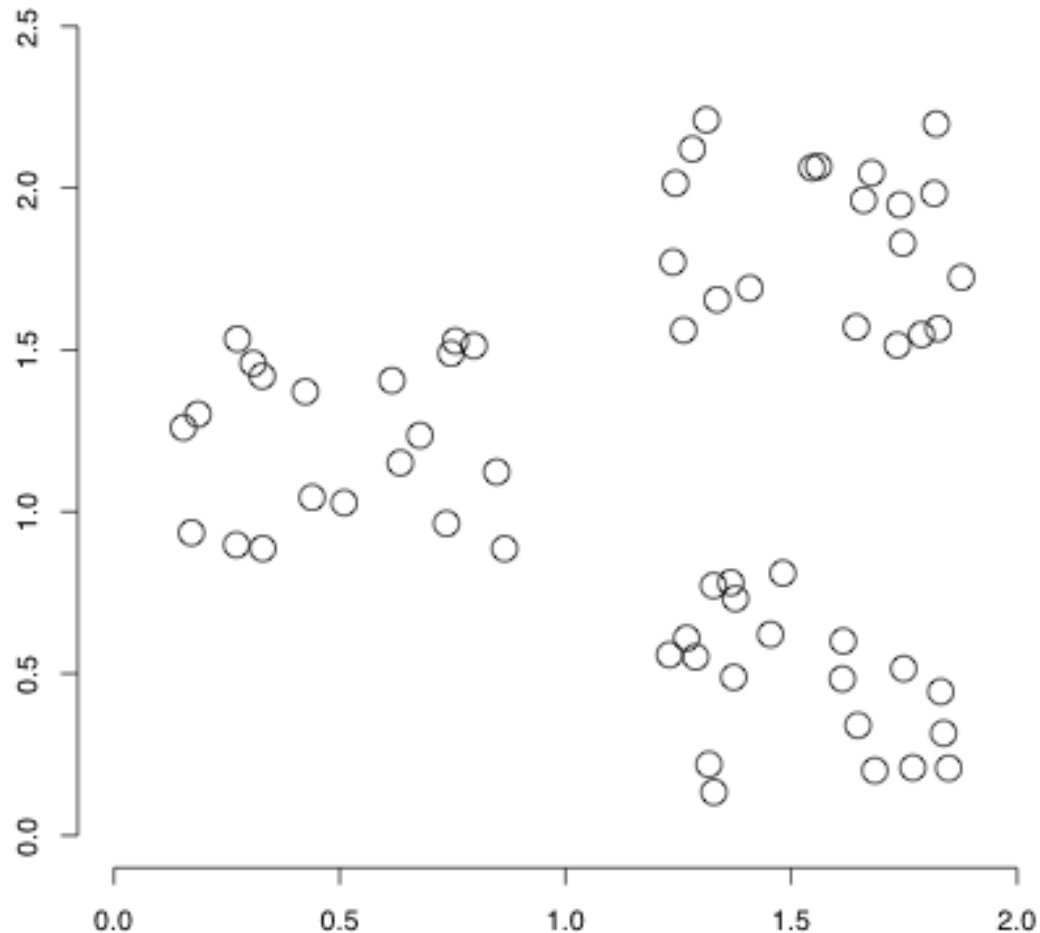
Text Clustering

Arzucan Özgür

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 a form of **unsupervised** learning.
- Unsupervised = there are no labeled or annotated data.

Data set with clear cluster structure



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: "closely associated documents tend to be relevant to the same requests".

Search result clustering for better navigation

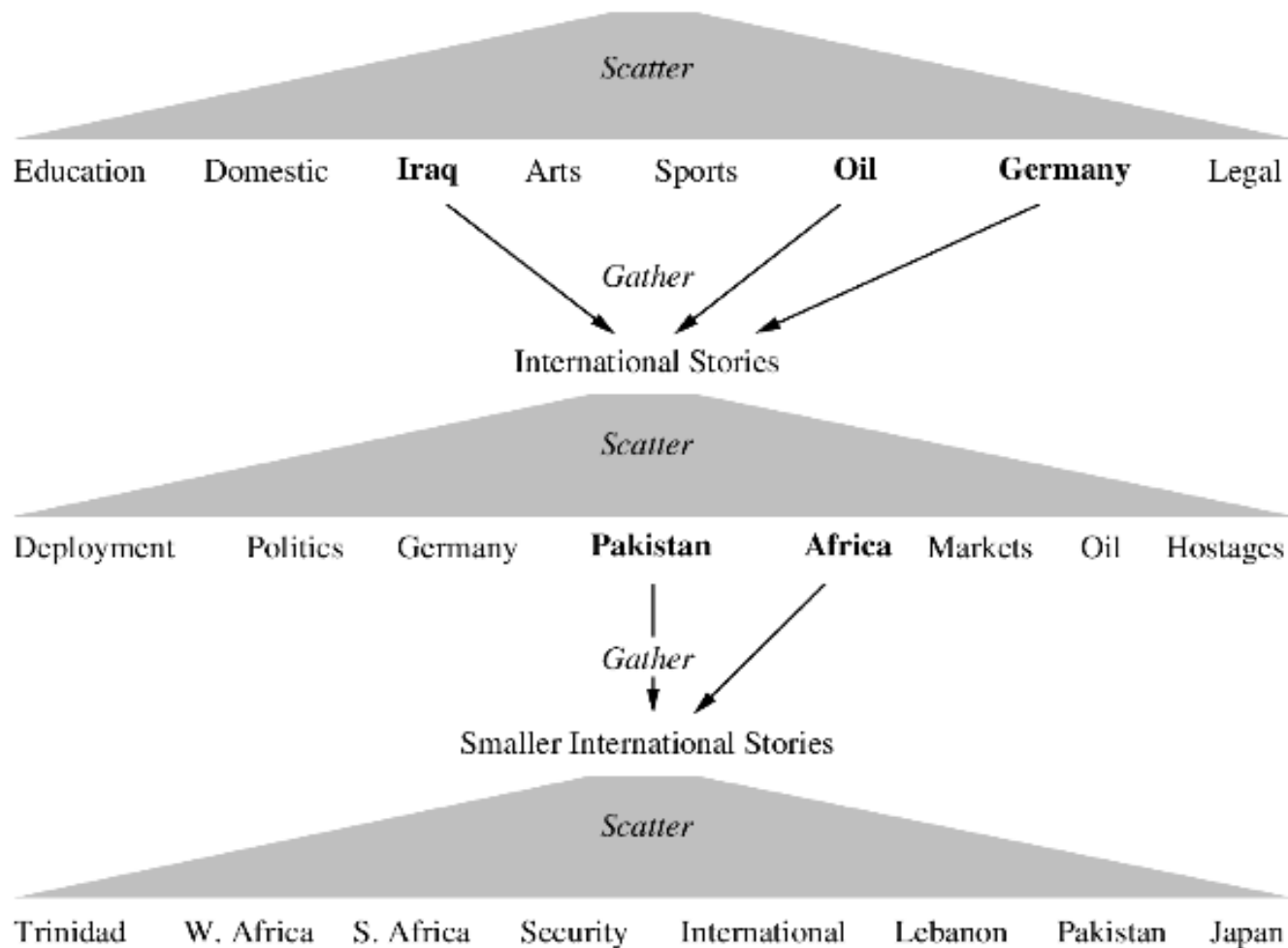


The screenshot displays the Vivísimo search engine interface. At the top, the Vivísimo logo is on the left, followed by a search bar containing the text 'jaguar' and a dropdown menu set to 'the Web'. To the right of the search bar is a blue 'Search' button and links for 'Advanced Search' and 'Help'. Below the search bar, a yellow banner indicates 'Top 200 results of at least 20,373,974 retrieved for the query **jaguar** (Details)'. On the left side, a 'Clustered Results' sidebar lists categories with expandable arrows and result counts: **jaguar** (208), **Cars** (74), **Club** (34), **Cat** (23), **Animal** (13), **Restoration** (10), **Mac OS X** (8), **Jaguar Model** (8), **Request** (5), **Mark Webber** (6), and **Maya** (5). A 'More' link is at the bottom of the sidebar. The main content area shows a list of search results:

- Jag-lovers - THE source for all Jaguar information** [new window] [frame] [cache] [preview] [clusters]
... Internet! Serving Enthusiasts since 1993 The Jag-lovers Web Currently with 40661 members The Premier **Jaguar** Cars web resource for all enthusiasts Lists and Forums Jag-lovers originally evolved around its ...
www.jag-lovers.org - Open Directory 2, Wisenut 8, Ask Jeeves 8, MSN 9, Looksmart 12, MSN Search 18
- Jaguar Cars** [new window] [frame] [cache] [preview] [clusters]
[...] redirected to www.jaguar.com
www.jaguarcars.com - Looksmart 1, MSN 2, Lycos 3, Wisenut 6, MSN Search 9, MSN 29
- <http://www.jaguar.com/>** [new window] [frame] [preview] [clusters]
www.jaguar.com - MSN 1, Ask Jeeves 1, MSN Search 3, Lycos 9
- Apple - Mac OS X** [new window] [frame] [preview] [clusters]
Learn about the new OS X Server, designed for the Internet, digital media and workgroup management. Download a technical factsheet.
www.apple.com/macosx - Wisenut 1, MSN 3, Looksmart 28

At the bottom left, there is a 'Find in clusters:' section with a text input field labeled 'Enter Keywords' and a red 'Go' button.

Scatter-Gather



Clustering for navigation: Google News

Google Haberler

http://news.google.com/

Most Visited Getting Started Latest Headlines Apple Amazon eBay Yahoo! News Arama Sonucu - sah...

Google Haberler

Web Görseller Haritalar Haberler Ceviri Akademik Gmail Diğer

Ayarlar | Oturum açın

Google haberler Türkiye

Haberleri Ara Web'de Ara

Gelişmiş haber arama

Türkiye

En Çok Okunan Haberler

Güncelleme: 7 dakika önce

En Çok Okunan Haberler

Yıldızlı ☆

Dünya

Türkiye

İş

Bilim/Teknoloji

Eğlence

Spor

Tüm içerikler

Başlıklar

Görseller

En Çok Okunan Haberler

Cihaner yargıya böyle direndi -Video ☆

Samanyolu Haber - 22 dakika önce

Hatırlayacaksınız aylar önce Cihaner'in gözaltı sırasında mahkeme heyeti ve savcılar eşkiyalıklı suçladığı görüntüleri yayınlamıştık. Cihaner o kayıta gelmem direnirim diyordu. İnanması güç ama savcı ünvanı taşıyan sanık dediğini yapmış ve kolluk ...

[İlhan Cihaner gelen polisleri direnmiş](#) TGRT Haber

[Cihaner'in gizli sorgu odası](#) Zaman

[Haber Türk](#) - [Hürriyet](#) - [Akşam](#) - [National Turk Haber](#)

[44 haber makalesinin tümü »](#) [E-postayla gönder](#)

AKP de CHP de askerin açıklamasını eleştirdi ☆

Vatan - 14 dakika önce

Meclis Başkanı Şahin'e göre açıklama yargıya müdahale olarak değerlendirildi. AKP'li Çelik ile CHP'li Hamzaçebi açıklamayı doğru bulmadıklarını söyledi. En sert tepki ise Kurtulmuş'tan geldi: Yargıya muhtıra ANKARA - Genelkurmay'ın "TSK'da görevli ve ...

[TSK'nın açıklamasına bir tepki de CHP'den](#) Haber Türk

[Chp'li Hamzaçebi: Genelkurmay'ın Açıklamasını Doğru Bulmadım \(1\)](#) Beyaz Gazete

[Ulusal Kanal](#) - [Haber X](#) - [Haberdar](#)

[65 haber makalesinin tümü »](#) [E-postayla gönder](#)

Bakan, GAP hedefini açıkladı

Samanyolu Haber - 1 Saat önce - [93 makalenin tümü »](#)

Japonya'da şiddetli deprem

Star Gazete - 1 Saat önce - [37 makalenin tümü »](#)

Fikret Hakan ağzını bozdu

Haber Türk - 10 saat önce - [28 makalenin tümü »](#)

İstanbul Büyükşehir Belediyespor: 0 - Gençlerbirliği: 1 (İlk yarı)

Zaman - 1 Saat önce - [82 makalenin tümü »](#)

Polise saldırıyı PKK üstlendi

Zaman - 3 saat önce - [117 makalenin tümü »](#)

Arıncı'tan basın özgürlüğü açıklaması

Sabah - 11 saat önce - [74 makalenin tümü »](#)

Euroda kritik soruya cevap geldi

Hürriyet - 5 saat önce - [141 makalenin tümü »](#)

Google, televizyona rakip olmak için kolları sıvadı

Bilgi Çağı - 7 saat önce - [22 makalenin tümü »](#)

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”.

Desiderata for clustering

- General goal: put related docs in the same cluster, put unrelated docs in different clusters.
 - How do we formalize 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 (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 sneakers in two clusters:
 - sports apparel
 - shoes
 - You can only do that with a soft clustering approach.
- We will do **flat, hard clustering, and hierarchical clustering** in this lecture.
- See Section 16.5 in the book for soft 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 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

K-means

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

Notion of similarity/distance

- ▶ Ideal: semantic similarity.
- ▶ Practical: term-statistical similarity
 - ▶ We will use cosine similarity.
 - ▶ Docs as vectors.
 - ▶ For many algorithms, easier to think in terms of a *distance* (rather than similarity) between docs.
 - ▶ We will mostly speak of Euclidean distance
 - ▶ But real implementations use cosine similarity



K-means

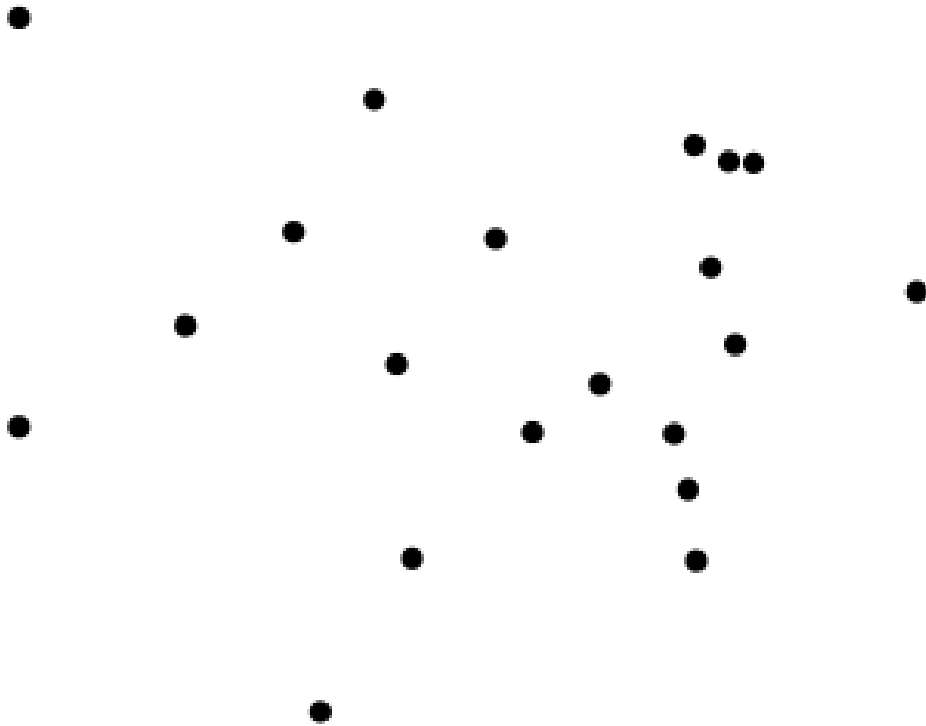
- Each cluster in K -means is defined by a **centroid**.
- Objective/partitioning criterion: **minimize the average squared difference from the centroid (RSS – Residual Squared Sum)**
- Recall definition of centroid:

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

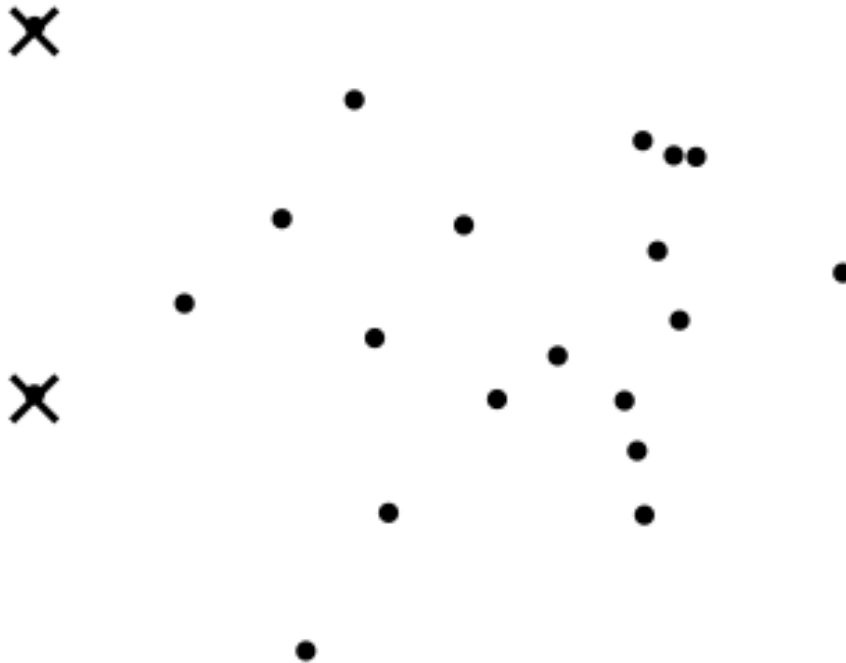
where we use ω to denote a cluster.

- 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

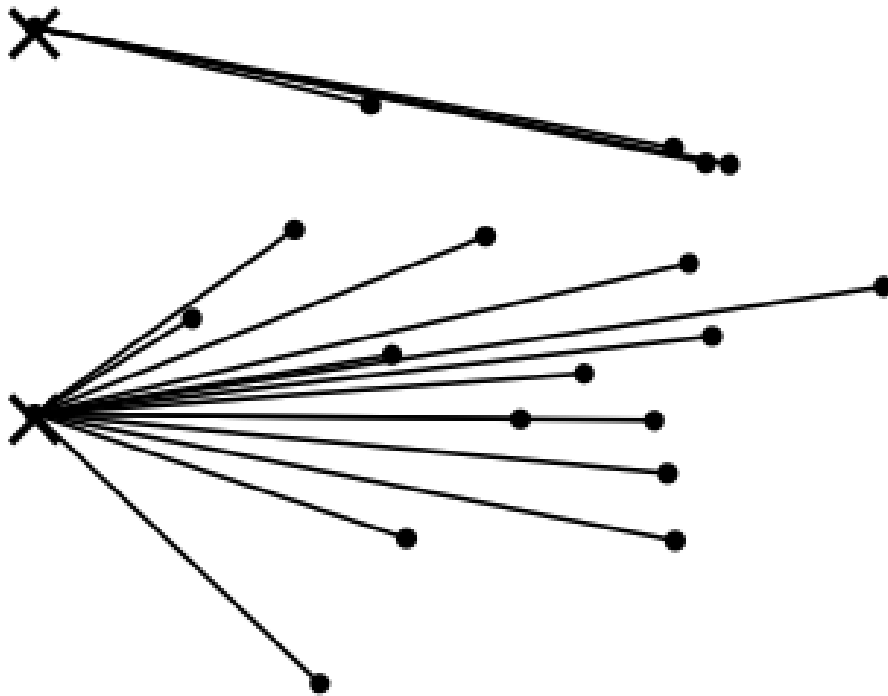
Worked Example: Set of to be clustered



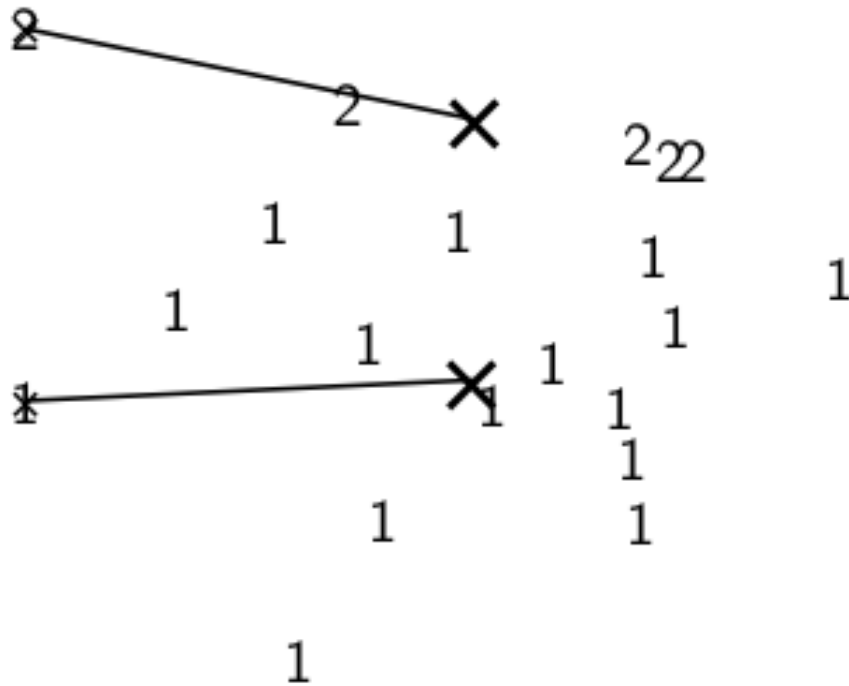
Worked Example: Random selection of initial centroids



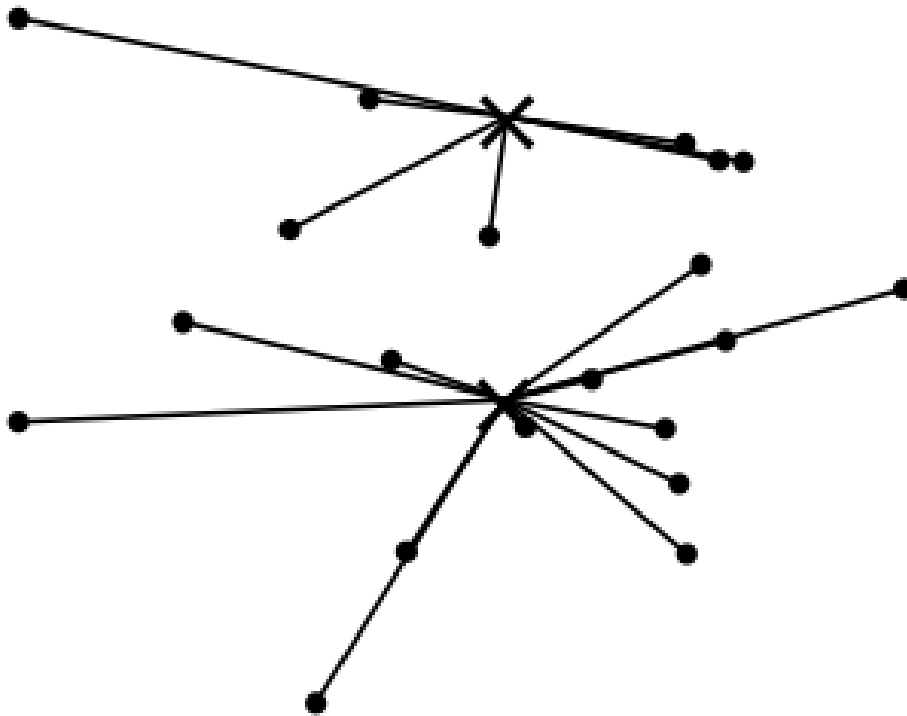
Worked Example: Assign points to closest center



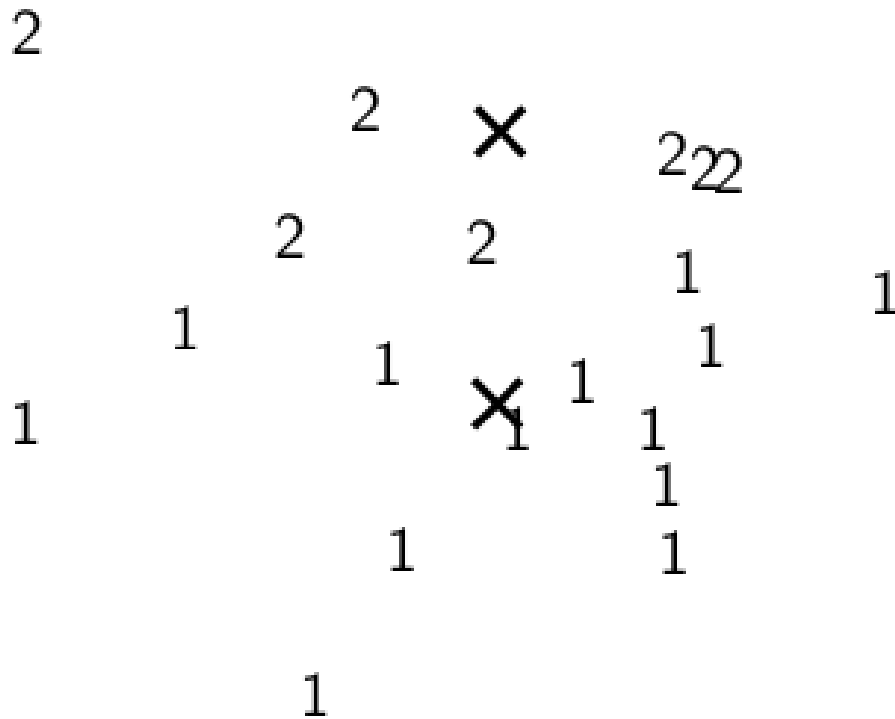
Worked Example: Recompute cluster centroids



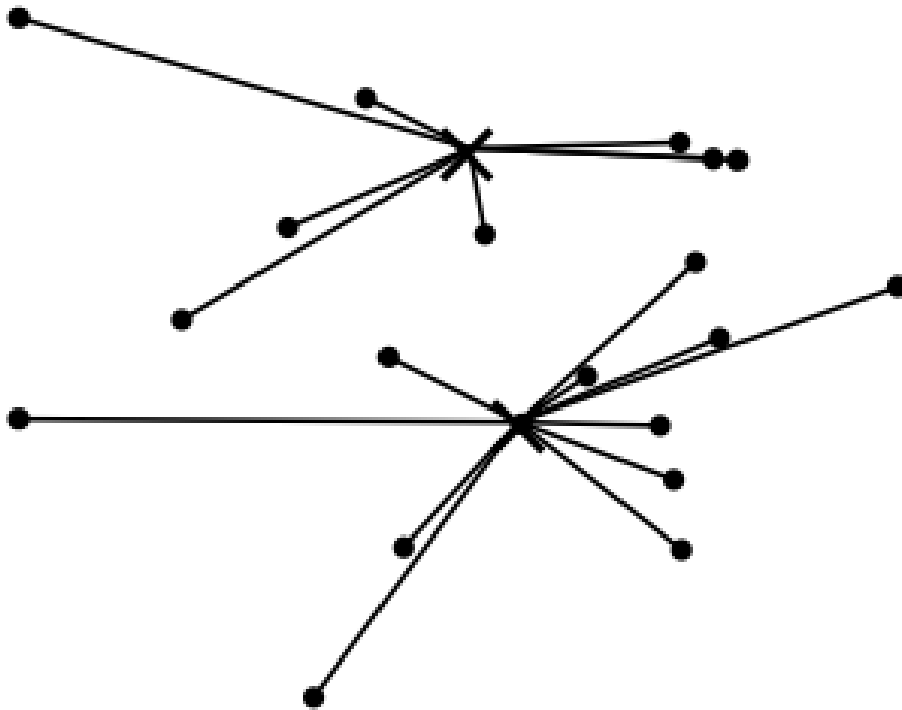
Worked Example: Assign points to closest centroid



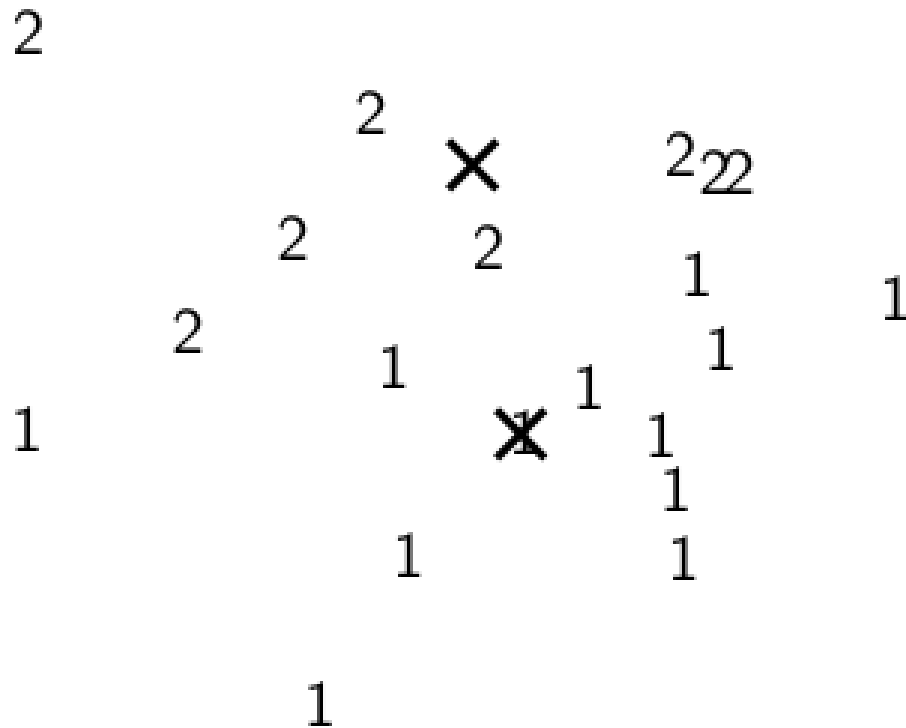
Worked Example: Assignment



Worked Example: Assign points to closest centroid

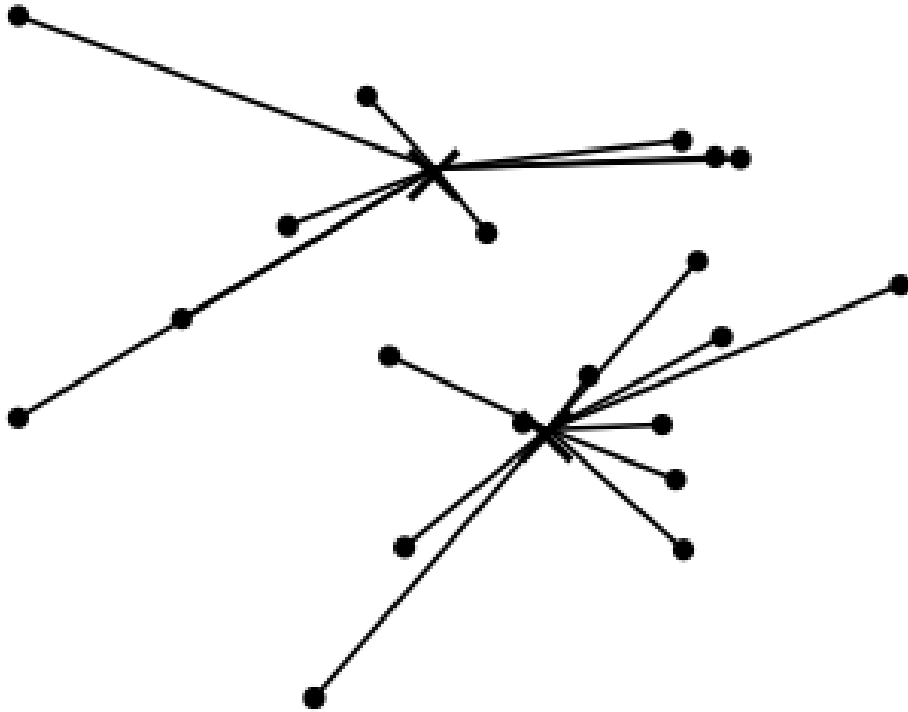


Worked Example: Assignment

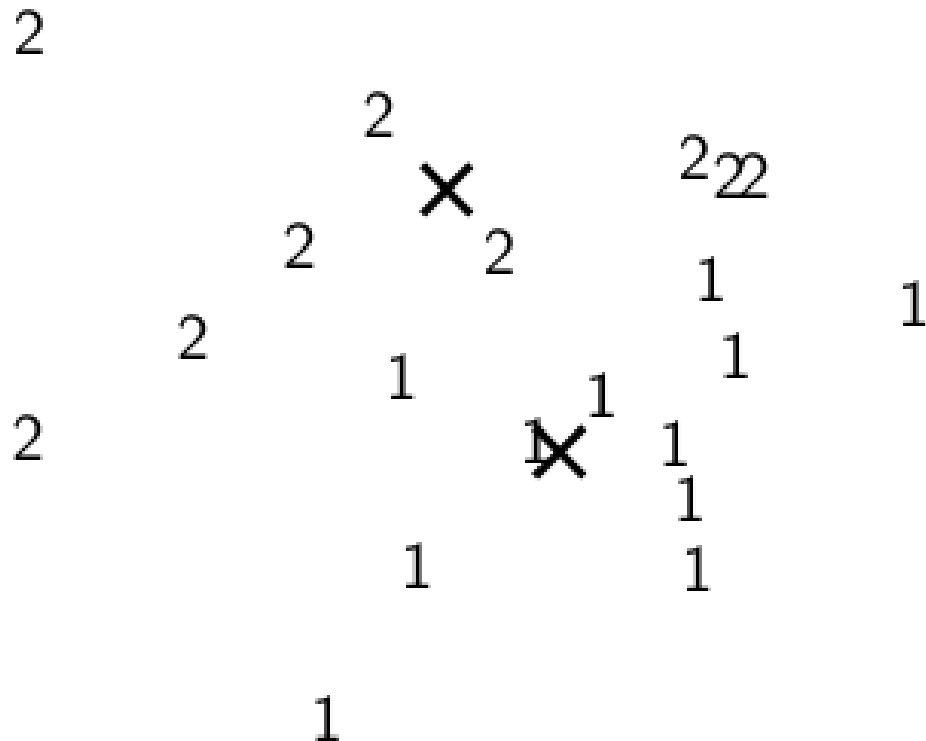




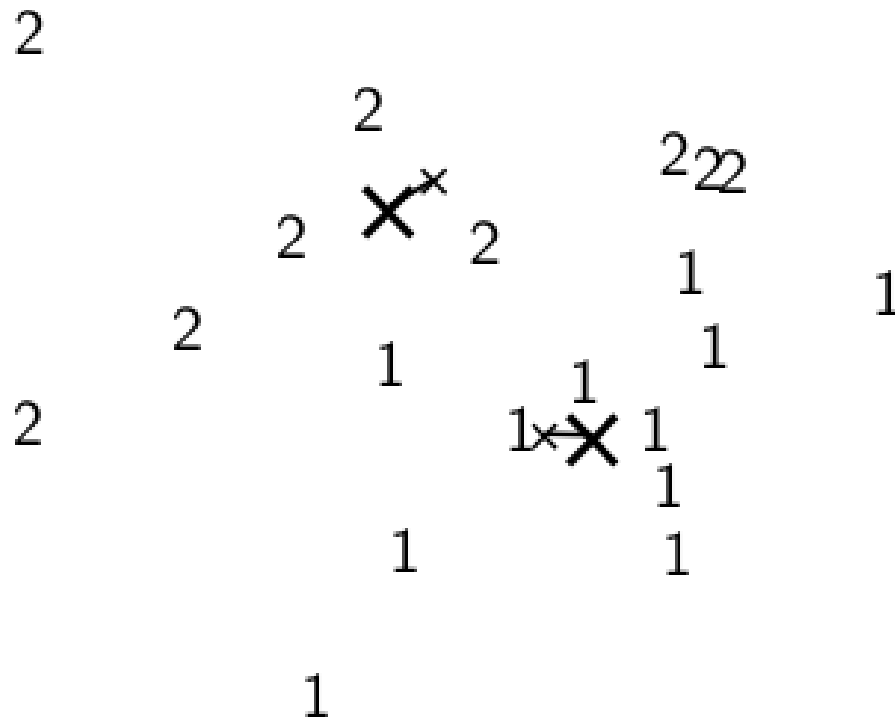
Worked Example: Assign points to closest centroid



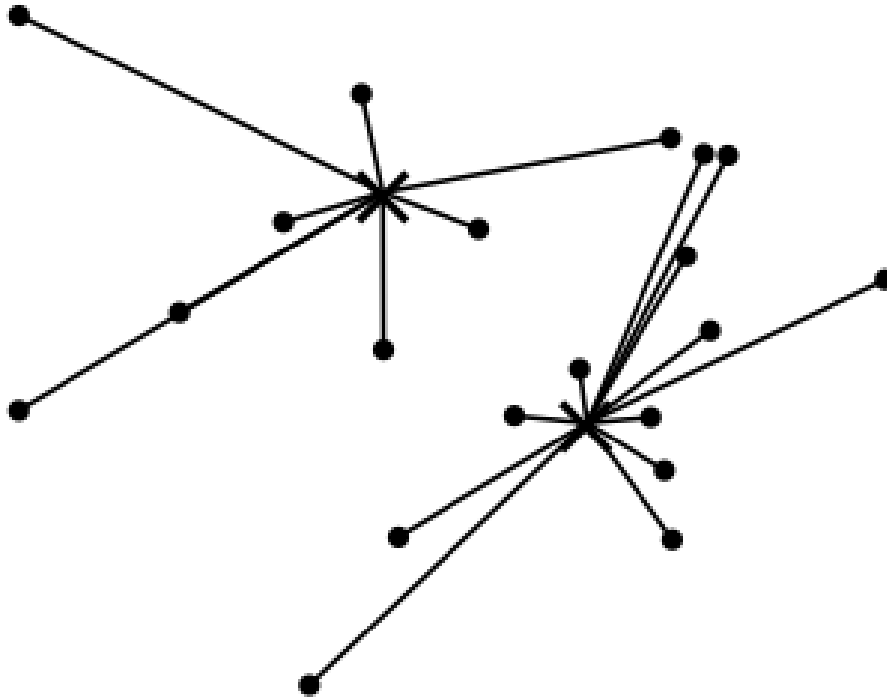
Worked Example: Assignment



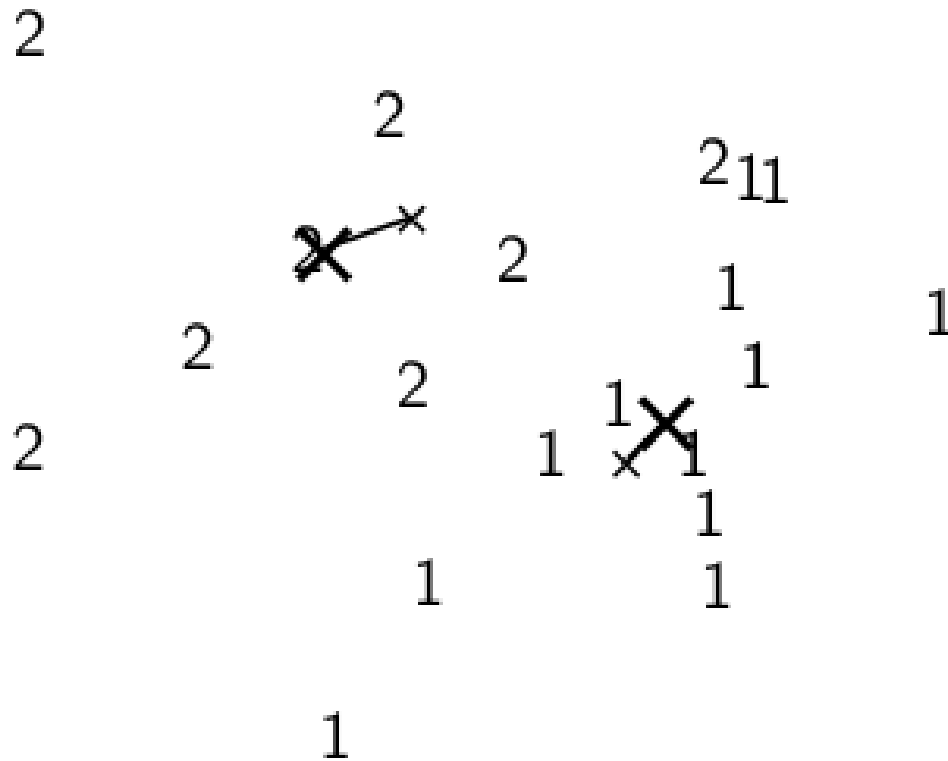
Worked Example: Recompute cluster centroids



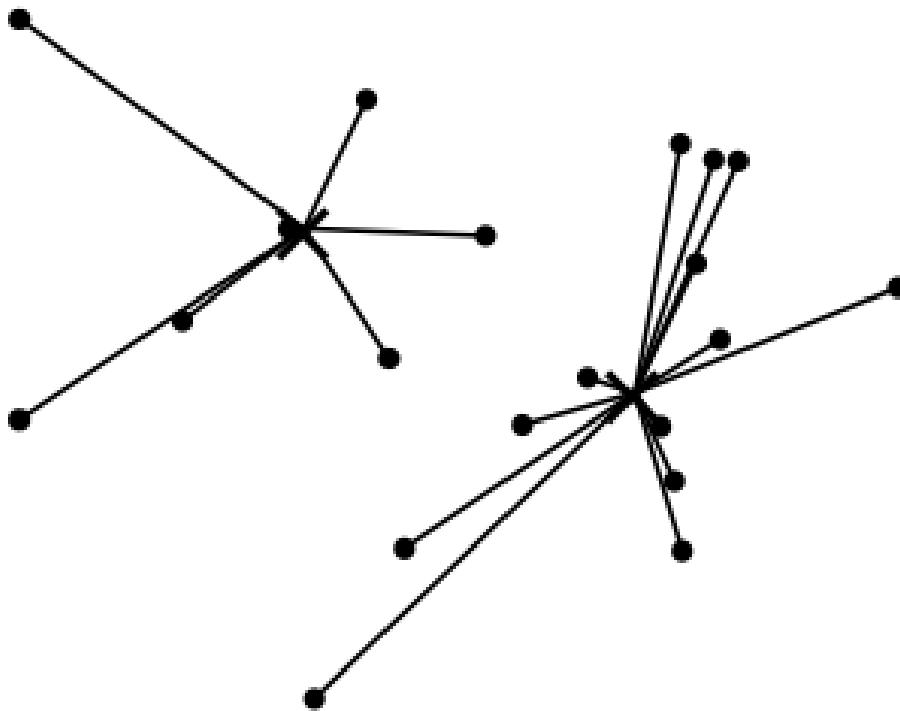
Worked Example: Assign points to closest centroid



Worked Example: Recompute cluster centroids

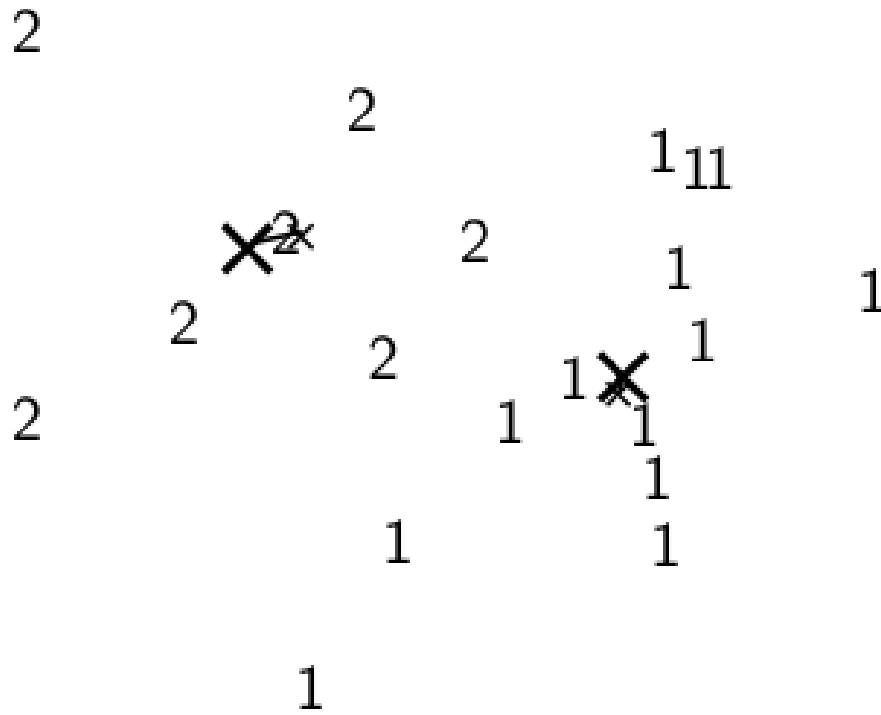


Worked Example: Assign points to closest centroid

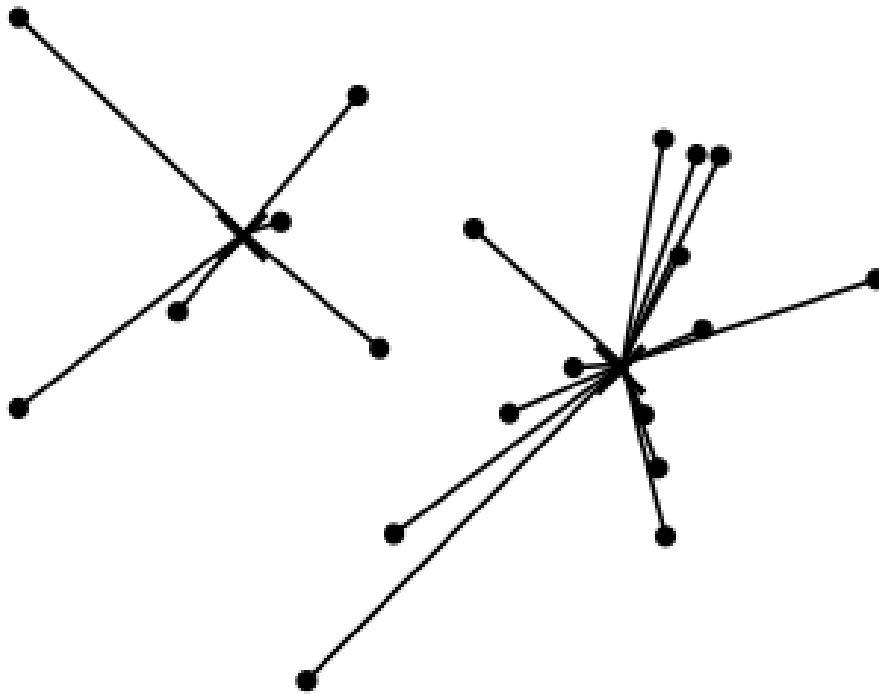




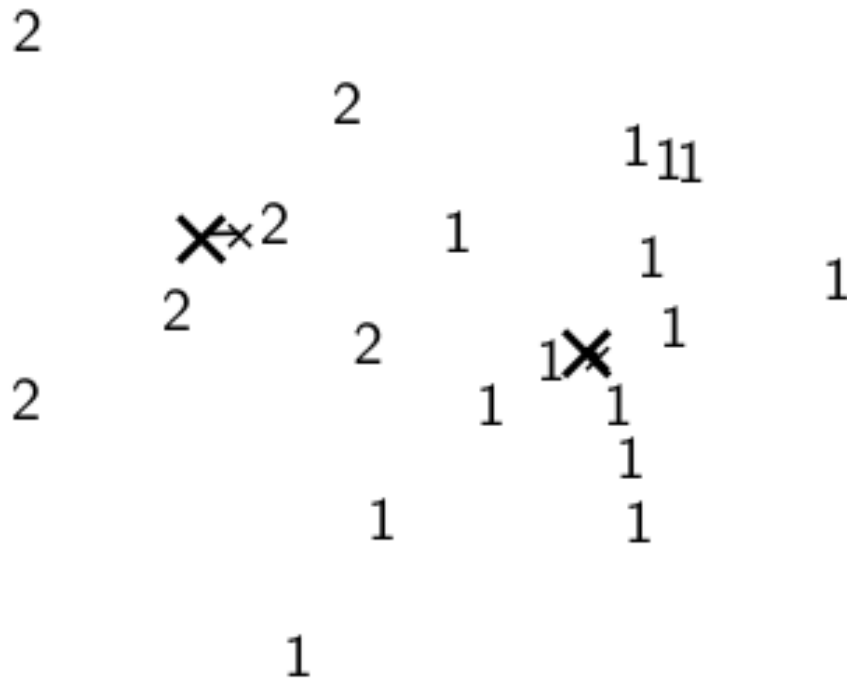
Worked Example: Recompute cluster centroids



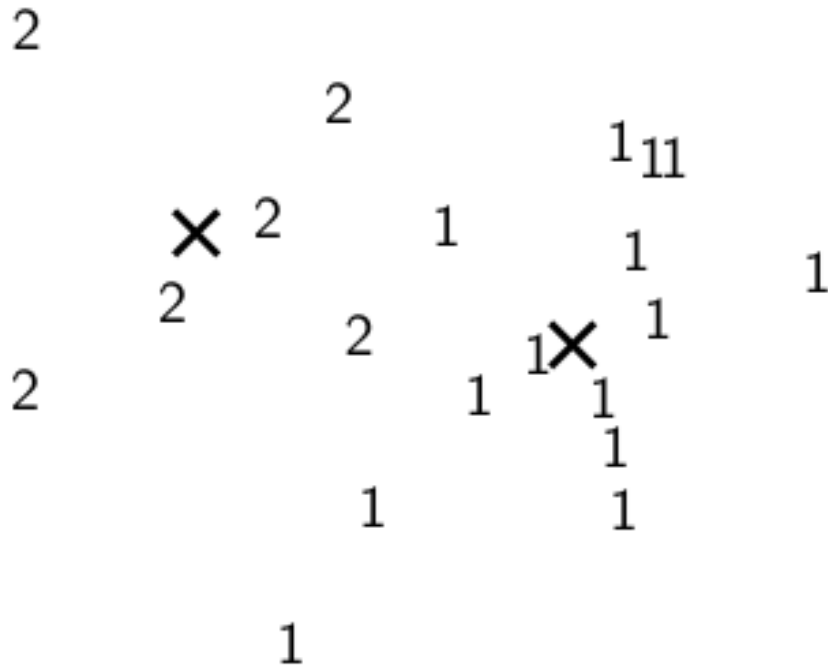
Worked Example: Assign points to closest centroid



Worked Example: Recompute cluster centroids



Worked Ex.: Centroids and assignments after convergence



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 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.

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

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

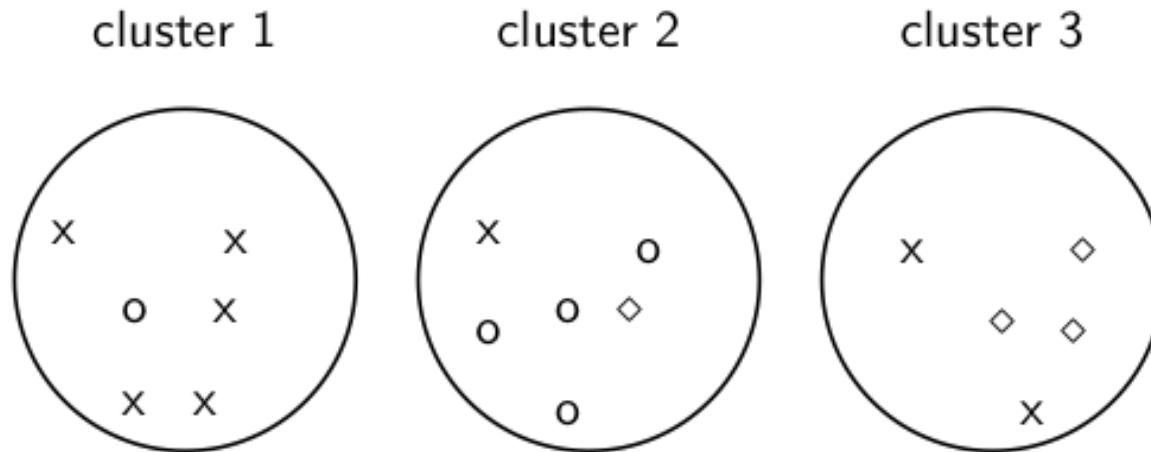
- Based on a gold standard data set, e.g., the Reuters collection we also used for the evaluation of classification
- Goal: Clustering should reproduce the classes in the gold standard
- (But we only want to reproduce how documents are divided into groups, not the class labels.)
- First measure for how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

External criterion: Purity

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

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



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

Rand index

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

■ 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)

... is the total number of pairs.

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

■ Example: $\binom{17}{2} = 136$ in o/◇/x example

■ Each pair is $\in \{0, \diamond, x\}$ or positive or negative (the clustering puts the two documents in the same or in different clusters) . . .

■ . . . and the clustering decision is correct or incorrect.

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:

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

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

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

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

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/◇/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).

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 (1)

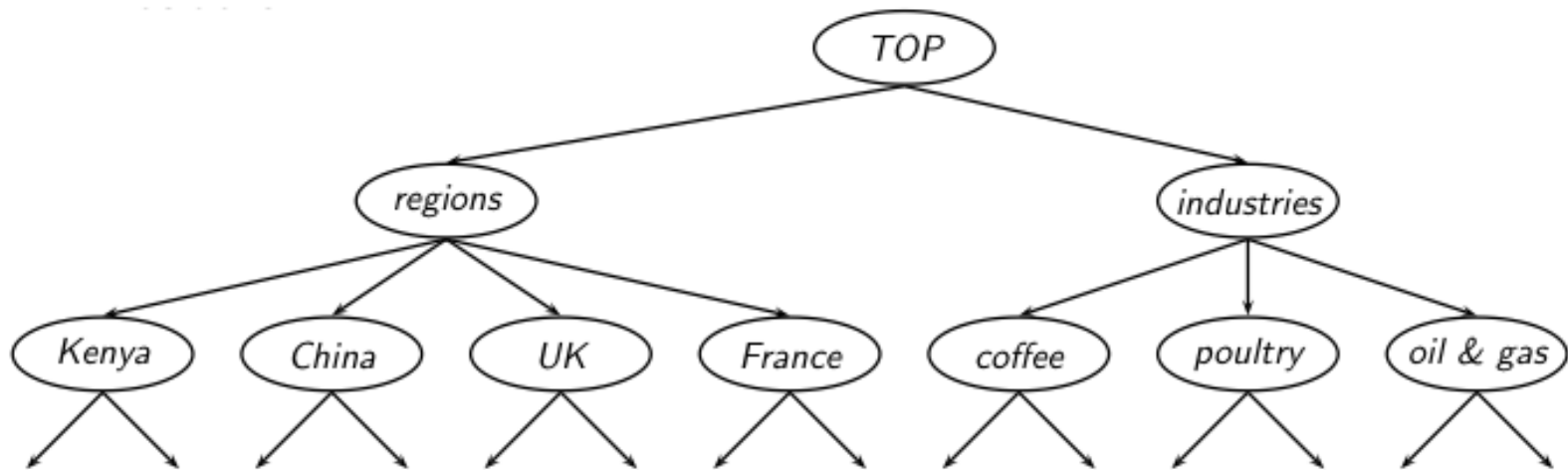
- Basic idea:
 - Start with 1 cluster ($K = 1$)
 - Keep adding clusters (= keep increasing K)
 - Add a penalty for each new cluster
- Trade off cluster penalties against average squared distance from centroid
- Choose the value of K with the best tradeoff

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 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 $\lambda \dots$

Hierarchical clustering

Our goal in hierarchical clustering is to create a hierarchy:

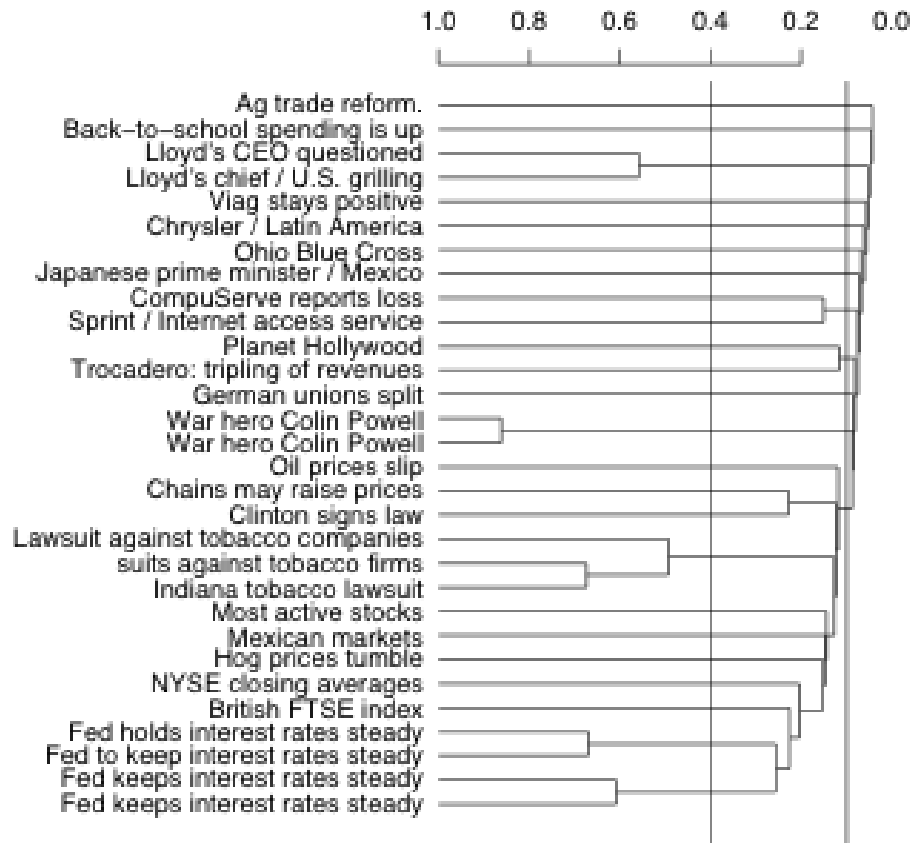


We want to create this hierarchy **automatically**. We can do this either **top-down** or **bottom-up**. The best known bottom-up method is **hierarchical agglomerative clustering**.

Hierarchical agglomerative clustering (HAC)

- Start with each document in a separate cluster
- Then repeatedly merge the two clusters that are most similar
- Until there is only one cluster
- The history of merging is a hierarchy in the form of a binary tree.
- The standard way of depicting this history is a dendrogram.

A dendrogram



- The history of mergers can be read off from bottom to top.
- The horizontal line of each merger tells us what the similarity of the merger was.
- We can cut the dendrogram at a particular point (e.g., at 0.1 or 0.4) to get a flat clustering.

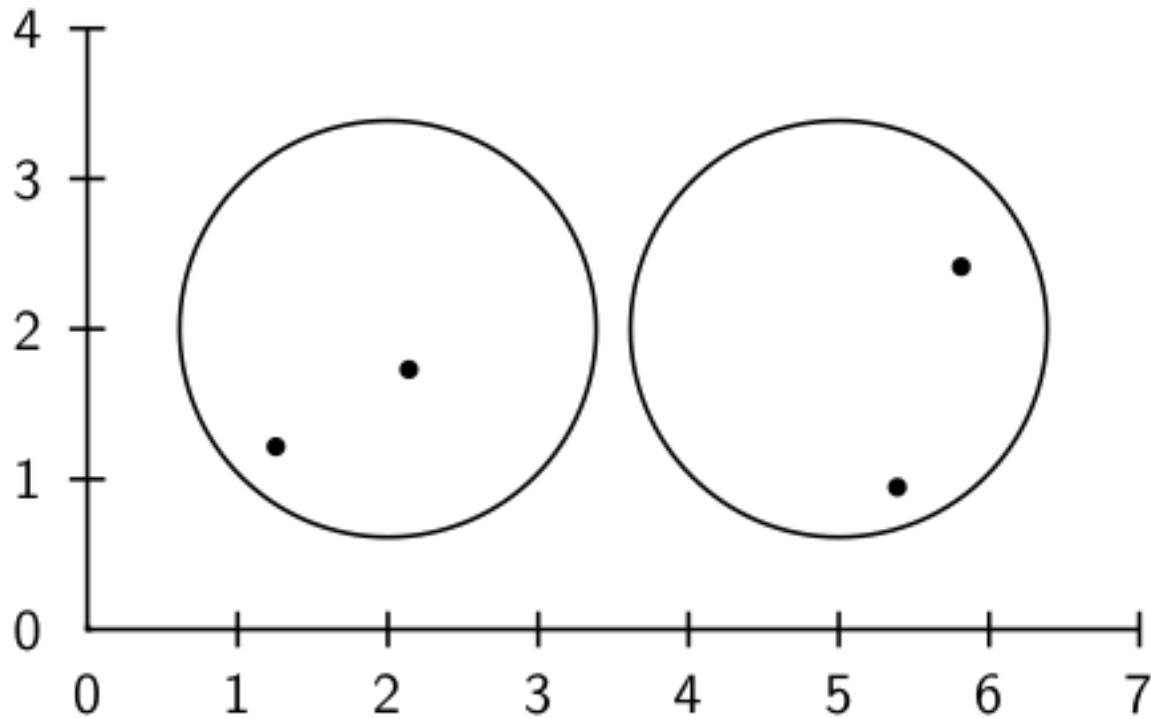
Divisive clustering

- Divisive clustering is top-down.
- Alternative to HAC (which is bottom up).
- Divisive clustering:
 - Start with all docs in one big cluster
 - Then recursively split clusters
 - Eventually each node forms a cluster on its own.
- → Bisecting K -means at the end
- For now: HAC (= bottom-up)

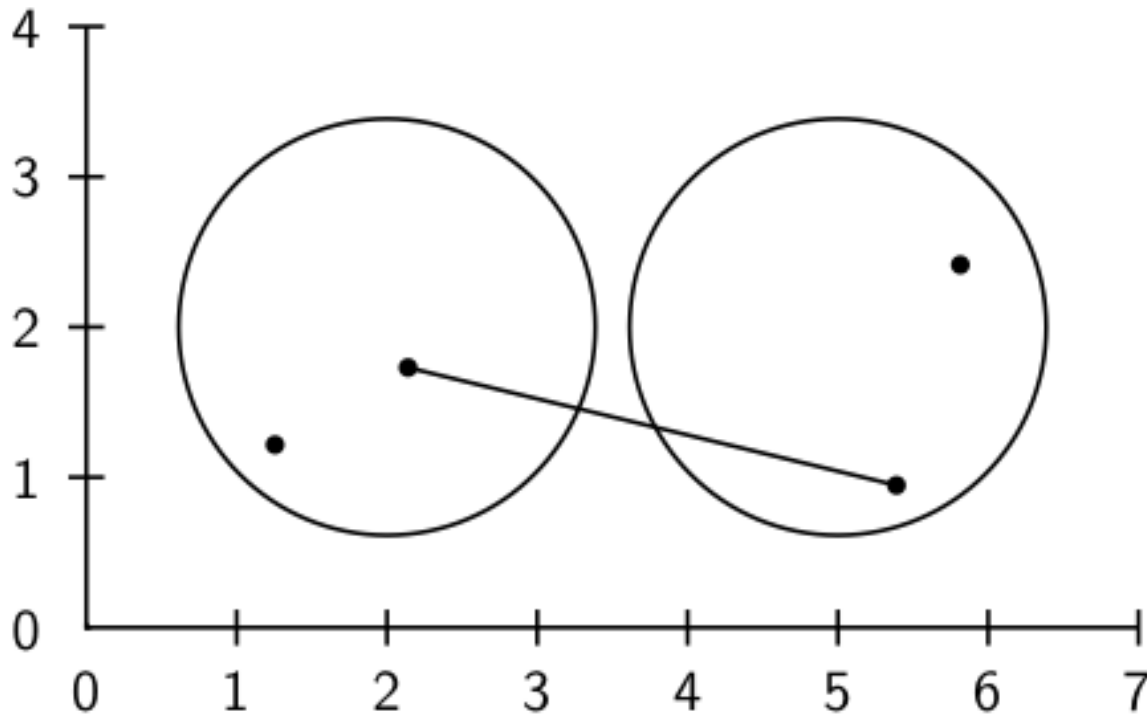
Key question: How to define cluster similarity

- Single-link: Maximum similarity
 - Maximum similarity of any two documents
- Complete-link: Minimum similarity
 - Minimum similarity of any two documents
- Centroid: Average “intersimilarity”
 - Average similarity of all document pairs (but excluding pairs of docs in the same cluster)
 - This is equivalent to the similarity of the centroids.
- Group-average: Average “intrasimilarity”
 - Average similarity of all document pairs, including pairs of docs in the same cluster

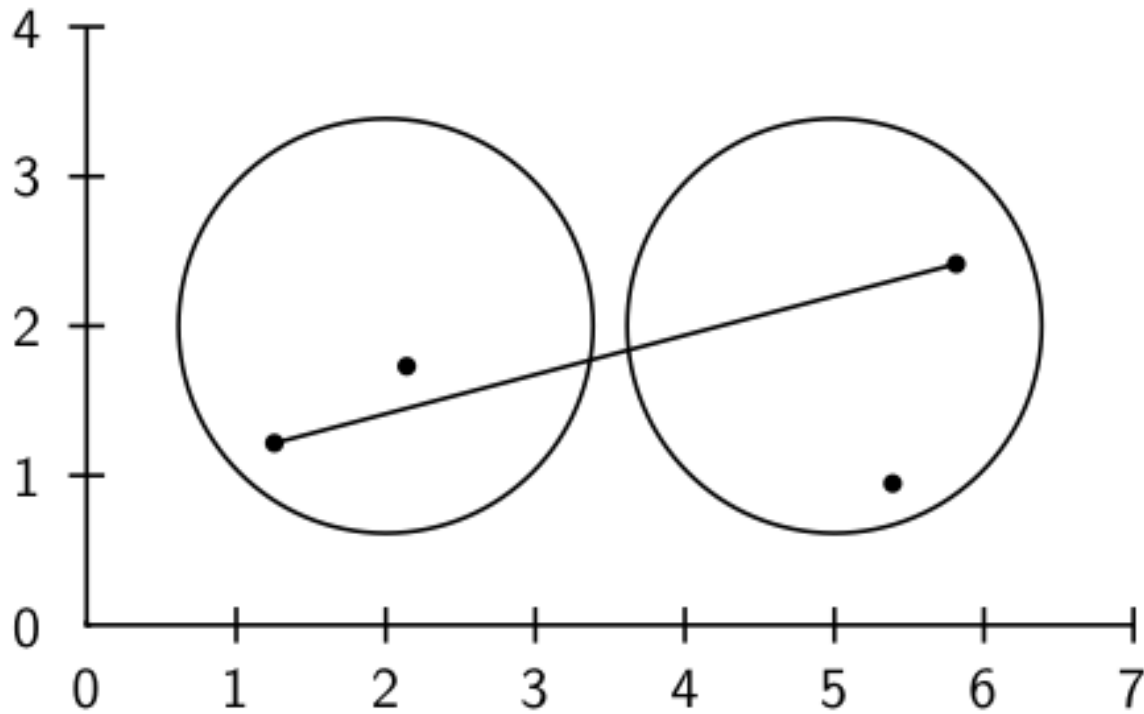
Cluster similarity: Example



Single-link: Maximum similarity

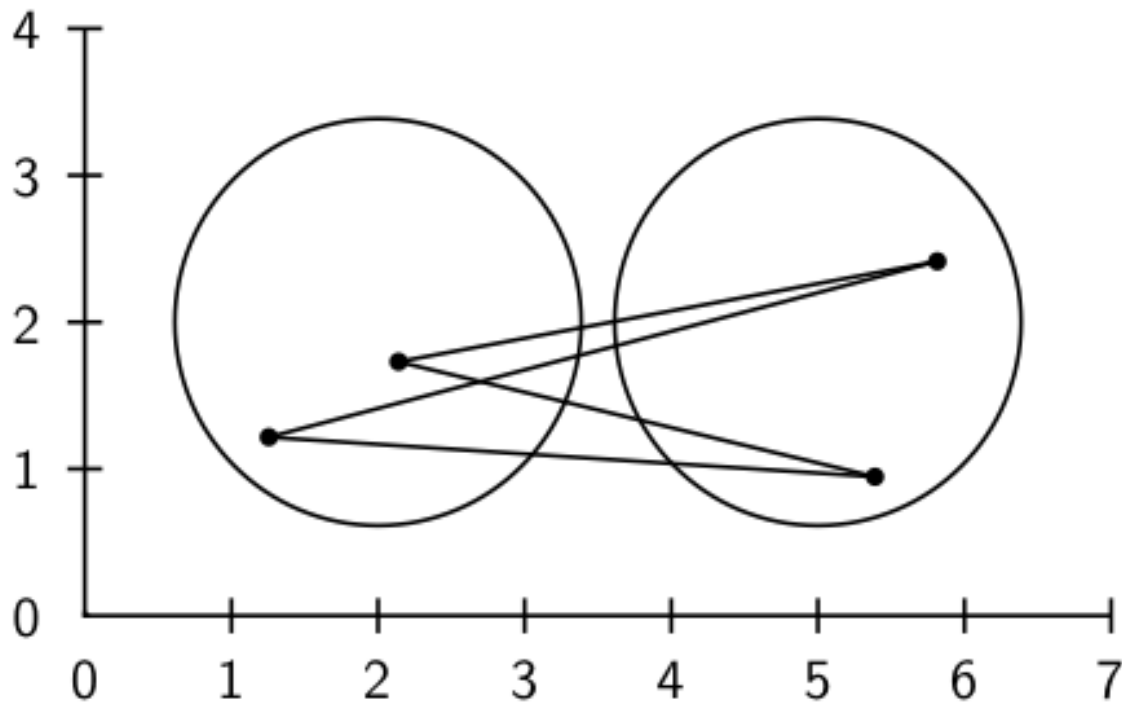


Complete-link: Minimum similarity



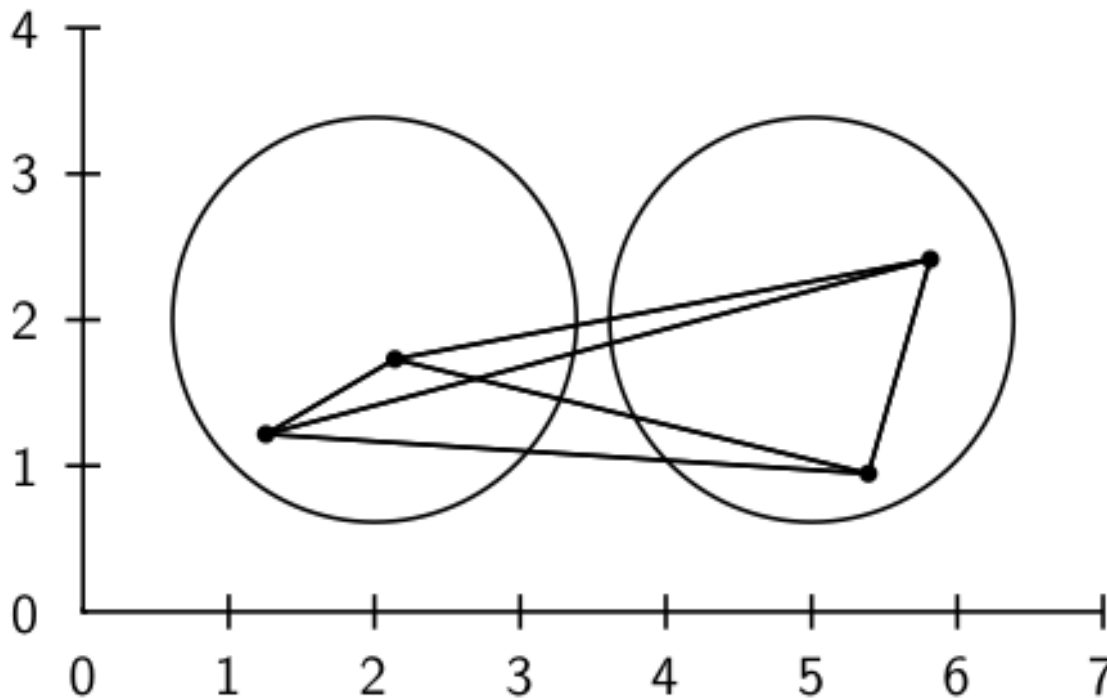
Centroid: Average intersimilarity

intersimilarity = similarity of two documents in different clusters

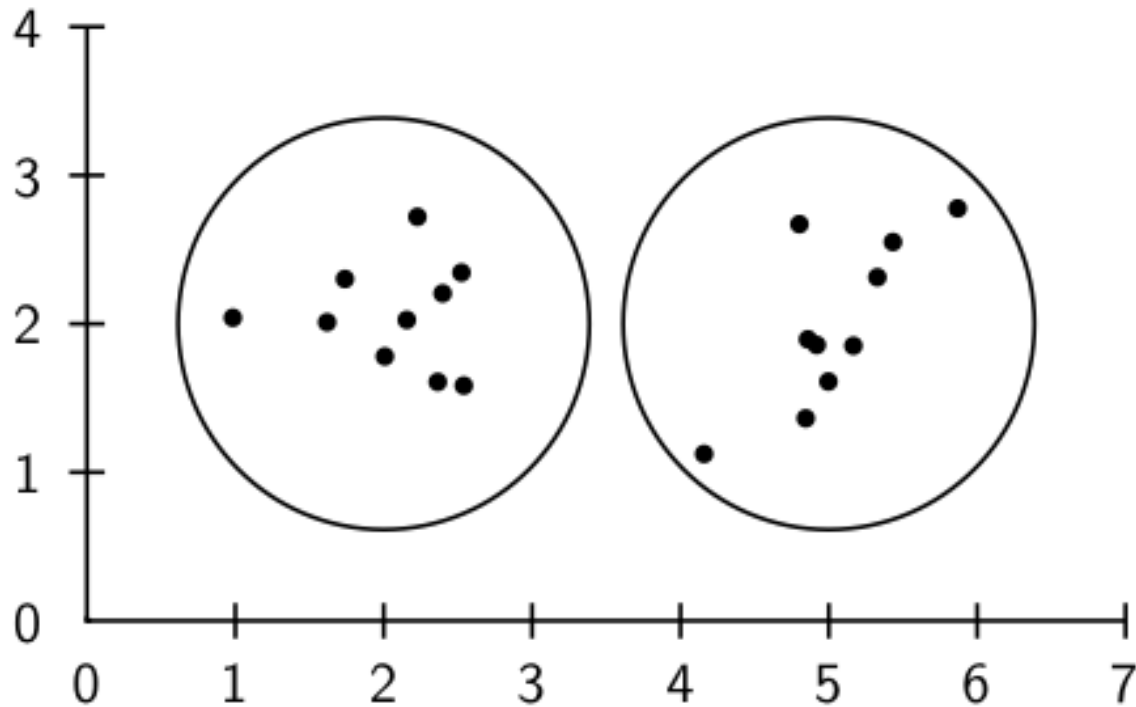


Group average: Average intrasimilarity

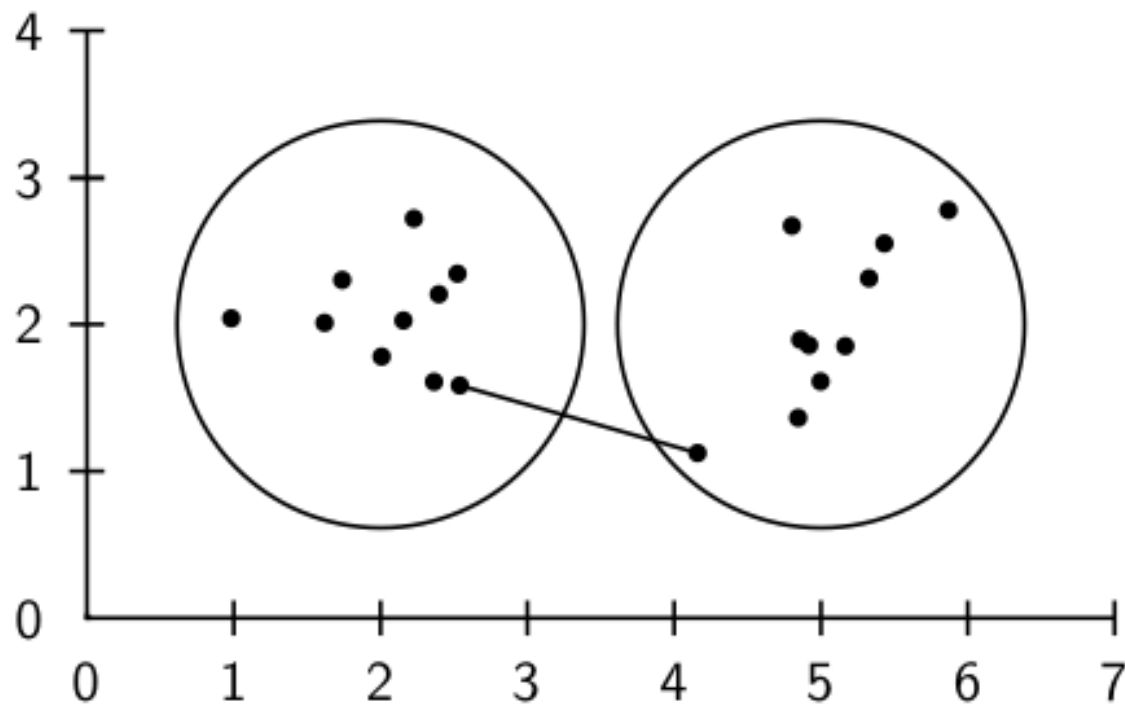
intrasimilarity = similarity of **any pair**, including cases where the two documents are in the same cluster



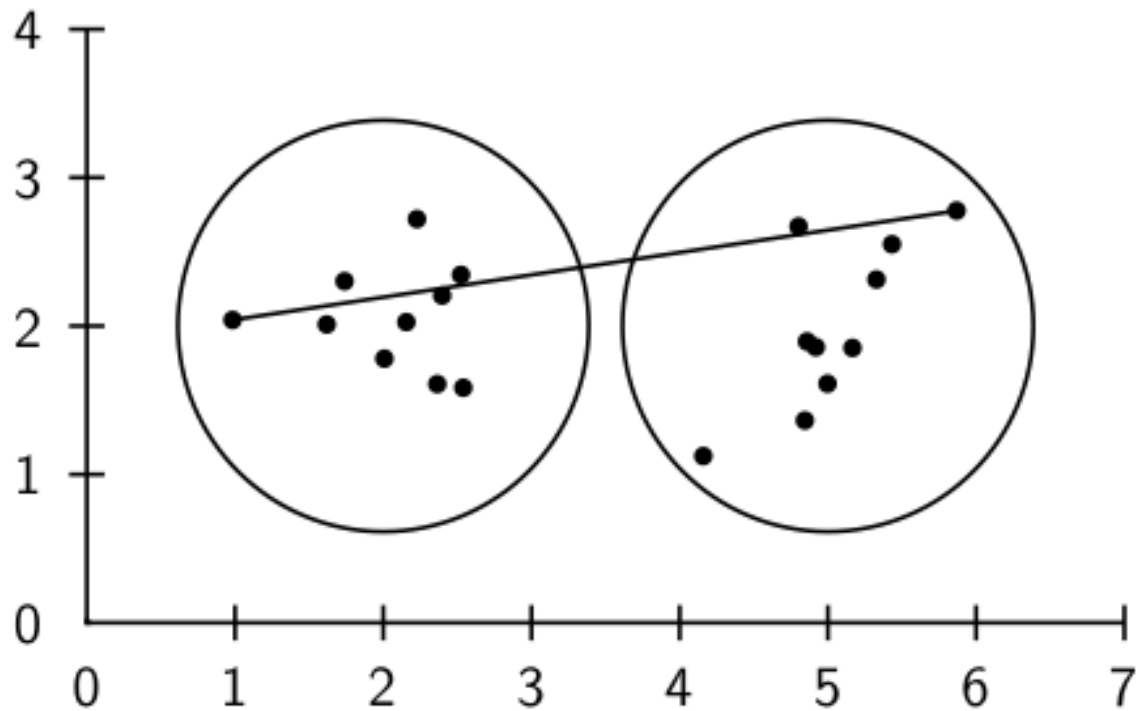
Cluster similarity: Larger Example



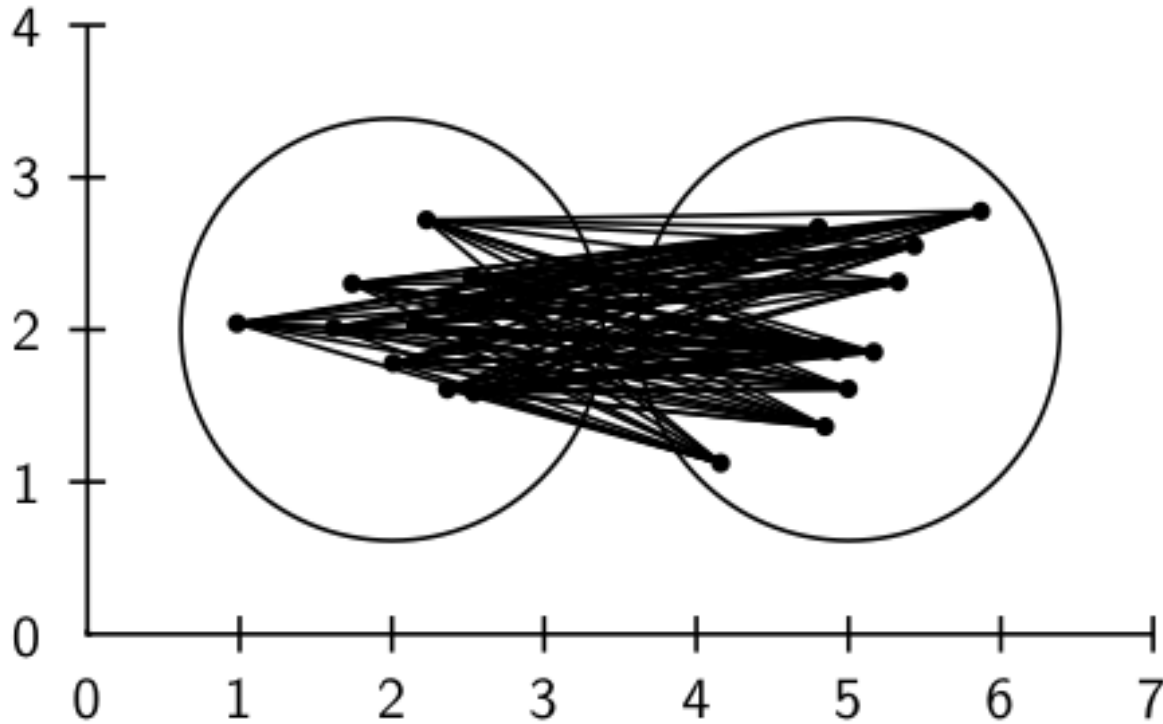
Single-link: Maximum similarity



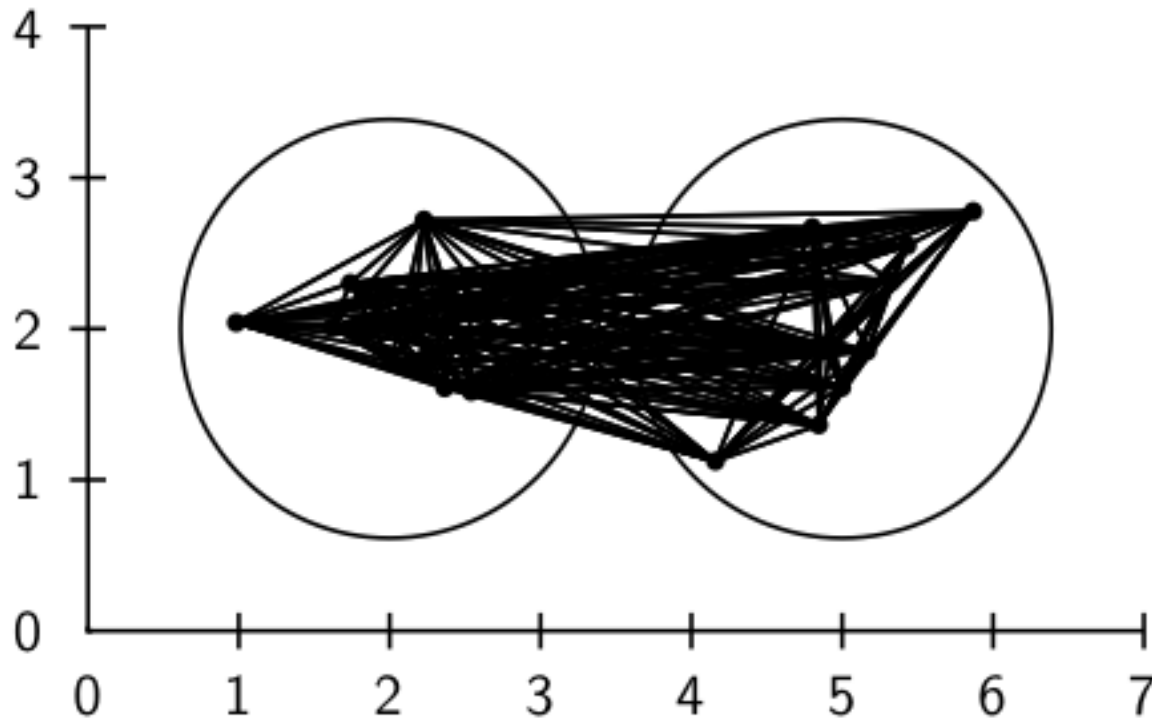
Complete-link: Minimum similarity



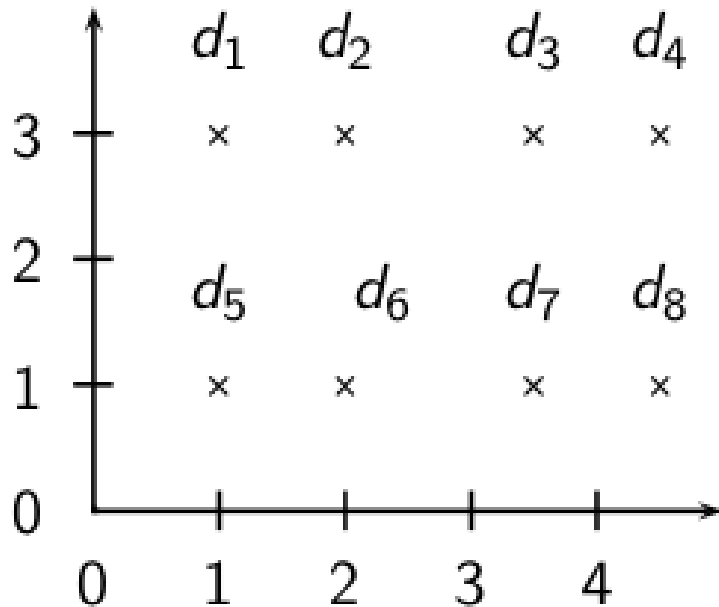
Centroid: Average intersimilarity



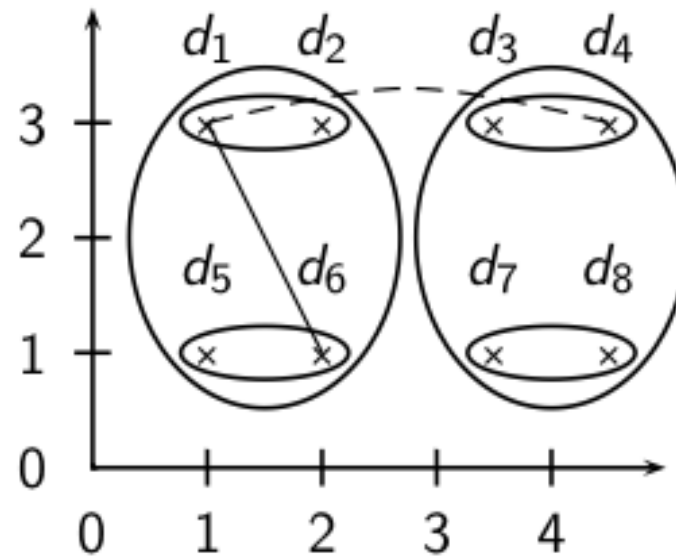
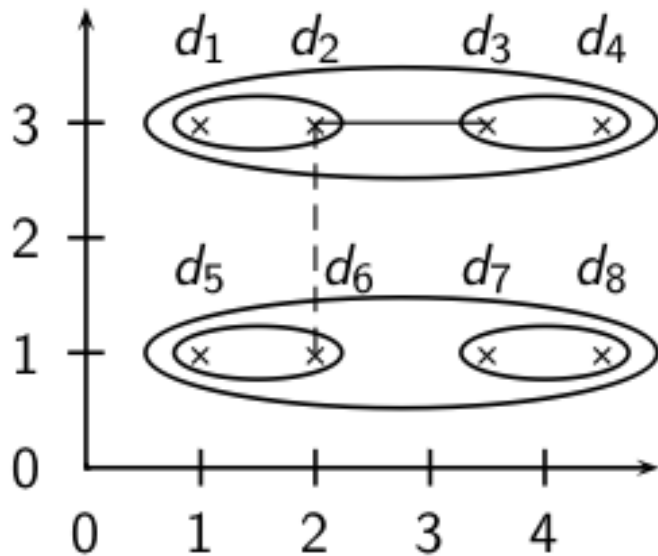
Group average: Average intrasimilarity



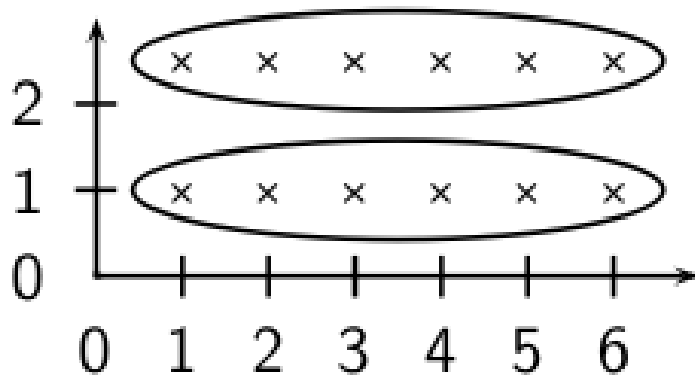
Exercise: Compute single and complete link clustering



Single-link vs. Complete link clustering

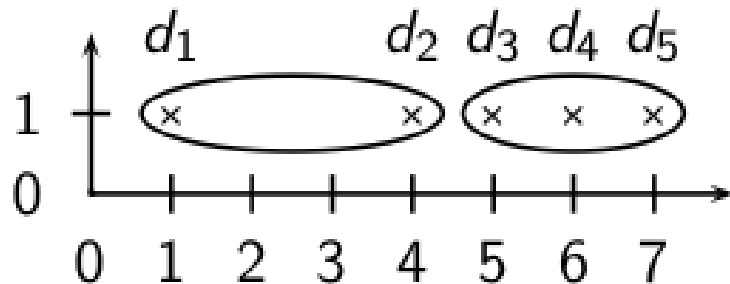


Single-link: Chaining



Single-link clustering often produces long, straggly clusters. For most applications, these are undesirable.

Complete-link: Sensitivity to outliers



- The complete-link clustering of this set splits d_2 from its right neighbors – clearly undesirable.
- The reason is the outlier d_1 .
- This shows that a single outlier can negatively affect the outcome of complete-link clustering.
- Single-link clustering does better in this case.

Which HAC clustering should I use?

- Don't use centroid HAC because of inversions.
- In most cases: GAAC is best since it isn't subject to chaining and sensitivity to outliers.
- However, we can only use GAAC for vector representations.
- For other types of document representations (or if only pairwise similarities for document are available): use complete-link.
- There are also some applications for single-link (e.g., duplicate detection in web search).

Flat or hierarchical clustering?

- For high efficiency, use flat clustering (or perhaps bisecting k -means)
- For deterministic results: HAC
- When a hierarchical structure is desired: hierarchical algorithm
- HAC also can be applied if K cannot be predetermined (can start without knowing K)

What to do with the hierarchy?

- Use as is (e.g., for browsing as in Yahoo hierarchy)
- Cut at a predetermined threshold
- Cut to get a predetermined number of clusters K
 - Ignores hierarchy below and above cutting line.

Bisecting K -means: A top-down algorithm

- Start with all documents in one cluster
- Split the cluster into 2 using K -means
- Of the clusters produced so far, select one to split (e.g. select the largest one)
- Repeat until we have produced the desired number of clusters

Major issue in clustering – labeling

- After a clustering algorithm finds a set of clusters: how can they be useful to the end user?
- We need a pithy label for each cluster.
- For example, in search result clustering for “jaguar”, The labels of the three clusters could be “animal”, “car”, and “operating system”.
- Topic of this section: How can we automatically find good labels for clusters?

Discriminative labeling

- To label cluster ω , compare ω with all other clusters
- Find terms or phrases that distinguish ω from the other clusters
- We can use any of the feature selection criteria we introduced in text classification to identify discriminating terms: mutual information, χ^2 , etc.

Non-discriminative labeling

- Select terms or phrases based solely on information from the cluster itself
- Terms with high weights in the centroid (if we are using a vector space model)
- Non-discriminative methods sometimes select frequent terms that do not distinguish clusters.
- For example, MONDAY, TUESDAY, . . . in newspaper text

Using titles for labeling clusters

- Terms and phrases are hard to scan and condense into a holistic idea of what the cluster is about.
- Alternative: titles
- For example, the titles of two or three documents that are closest to the centroid.
- Titles are easier to scan than a list of phrases.

Cluster labeling: Example

	# docs	labeling method		
		centroid	mutual information	title
4	622	oil plant mexico production crude power 000 refinery gas bpd	plant oil production barrels crude bpd mexico dolly capacity petroleum	MEXICO: Hurricane Dolly heads for Mexico coast
9	1017	police security russian people military peace killed told grozny court	police killed military security peace told troops forces rebels people	RUSSIA: Russia's Lebed meets rebel chief in Chechnya
10	1259	00 000 tonnes traders futures wheat prices cents september tonne	delivery traders futures tonne tonnes desk wheat prices 000 00	USA: Export Business - Grain/oilseeds complex

- Three methods: most prominent terms in centroid, differential labeling using MI, title of doc closest to centroid
- All three methods do a pretty good job.

Resources

- ▶ *Introduction to Information Retrieval*, chapters 16,17.
- ▶ Some slides were adapted from
 - ▶ Prof. Dragomir Radev's lectures at the University of Michigan:
 - ▶ <http://clair.si.umich.edu/~radev/teaching.html>
 - ▶ the book's companion website:
 - ▶ <http://nlp.stanford.edu/IR-book/information-retrieval-book.html>
- ▶ Weka: A data mining software package that includes an implementation of many Machine Learning algorithms

