Clustering

IIR secs 16

SAPIENZA
UNIVERSITÀ DI ROMA

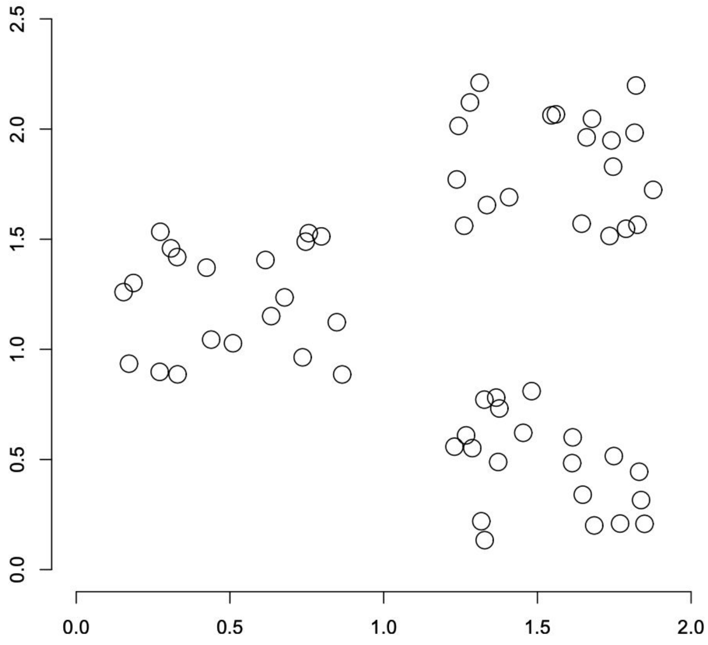
Fabrizio Silvestri

Recap and Quick Intro

Learning to Rank

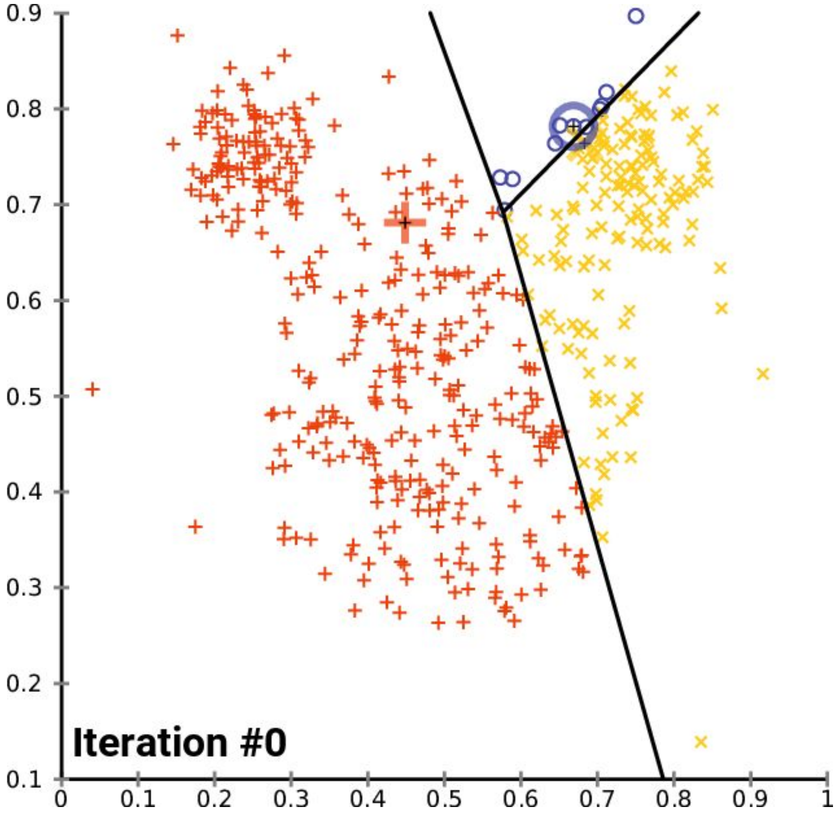
- The problem of making a binary relevant/nonrelevant judgment is cast as a classification or regression problem, based on a training set of query-document pairs and associated relevance judgments.
- In principle, any method learning a classifier (including least squares regression) can be used to find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

An Example



Propose algorithm for finding the cluster structure in this example □

What's Clustering?



Clustering

Clustering: Definitions

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

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, . . .

Clustering in IR

Classification Vs. Clustering

Cluster hypothesis.

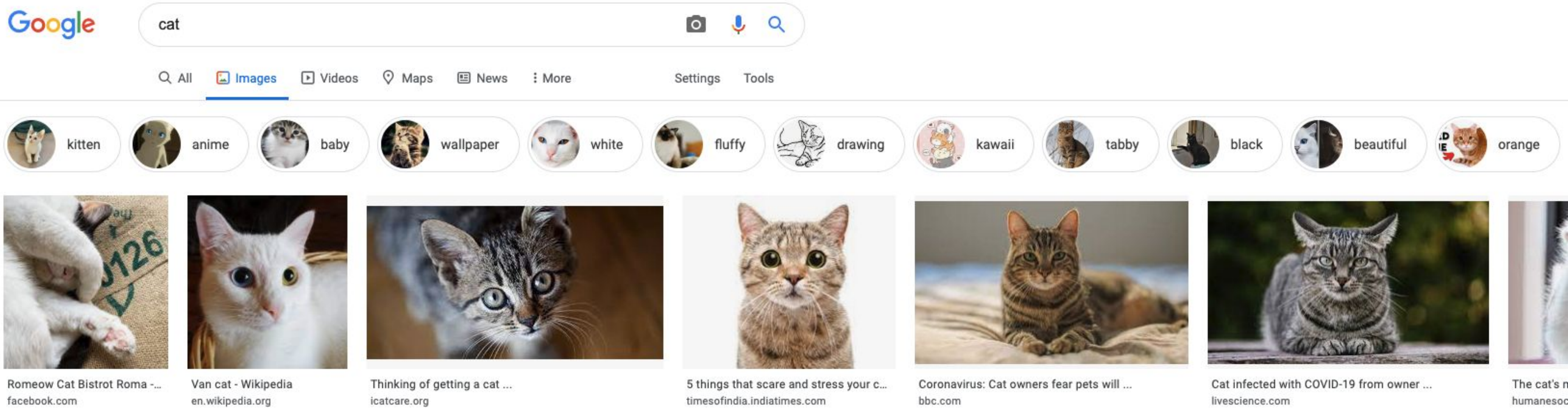
Documents in the same cluster behave similarly with respect to relevance to information needs.

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 in IR

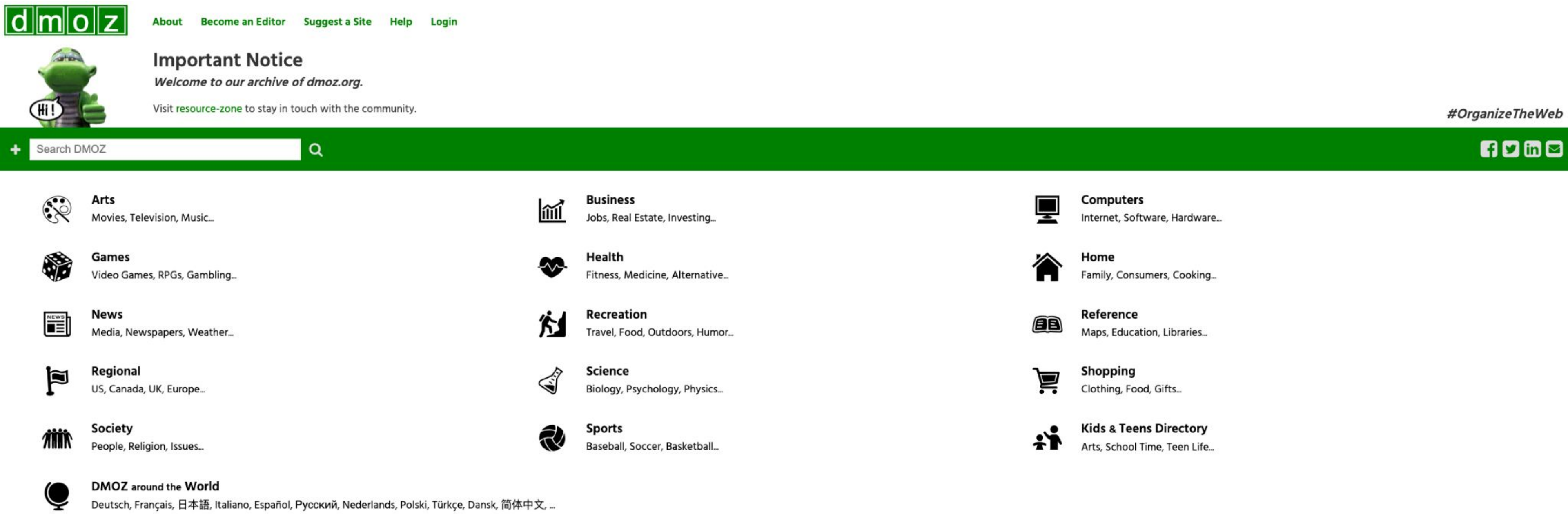
application	what is clustered?	benefit
search result clustering	search results	more effective information presentation to user
Scatter-Gather	(subsets of) collection	alternative user interface: "search without typing"
collection clustering	collection	effective information presentation for exploratory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

Applications in IR



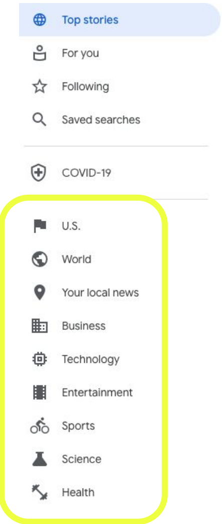
SAPIENZA
UNIVERSITÀ DI ROMA

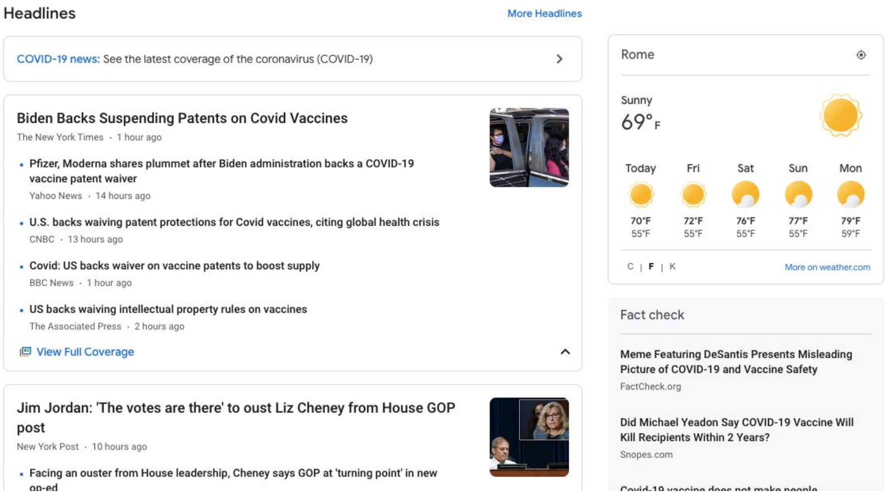
Applications in IR



How do you build directories?

- DMOZ is manually built…
- Can you do it automatically?
    - E.g. Google News





Can Clustering Improve 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”.

Let’s go Deeper

What clustering should do?

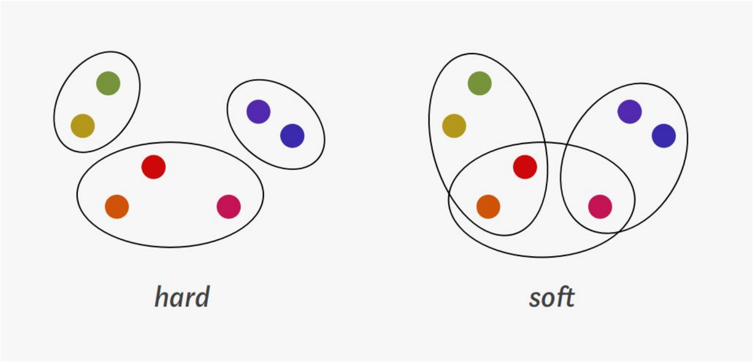
- General goal: put related docs in the same cluster, put unrelated docs in different clusters.
  - We'll see different ways of formalizing this.
- The number of clusters should be appropriate for the data set we are clustering.
  - Initially, we will assume the number of clusters K is given.
  - Later: 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 (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, and shoes
  - You can only do that with a soft clustering approach.



Flat Clustering

- 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

Axioms for Clustering

- Invariance
  - Clustering should not depend on how we measure distances (e.g., feet, metres, inches, etc.)
- Richness
  - Clustering induces a partition on the set of objects we are partitioning. A good clustering algorithm should not rule out any partition a-priori.
- Consistency
  - Once objects are partitioned into clusters, reducing the distance between objects in a cluster or increasing the distance between objects in different clusters should not impact the clustering result

Impossibility Theorem for Clustering

- John Kleimberg. “An Impossibility Theorem for Clustering”. NIPS 2015

Theorem 2.1 *For each $n \geq 2$, there is no clustering function f that satisfies Scale-Invariance, Richness, and Consistency.*

K-Means

What’s K-Means?

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

How are documents represented?

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

K-Means: Basic Idea

- 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:

cluster

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

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

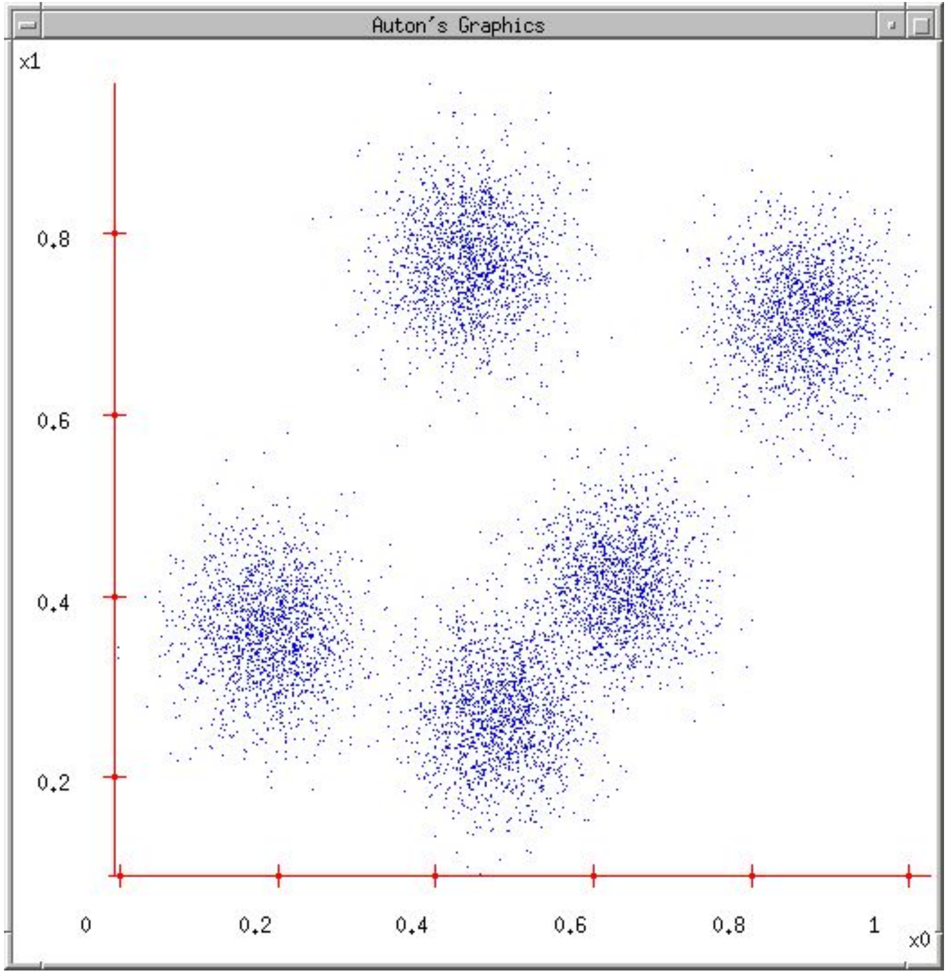
K-Means: pseudocode (μ_k is centroid of ω_k)

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

$(\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_K) \leftarrow$ SELECTRANDOMSEEDS$(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$
for $k \leftarrow 1$ **to** K
do $\vec{\mu}_k \leftarrow \vec{s}_k$
while stopping criterion has not been met
do for $k \leftarrow 1$ **to** K
    do $\omega_k \leftarrow \{\}$
    for $n \leftarrow 1$ **to** N
    do $j \leftarrow \arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$
        $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ *(reassignment of vectors)*
    for $k \leftarrow 1$ **to** K
    do $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$ *(recomputation of centroids)*
return $\{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$

K-means

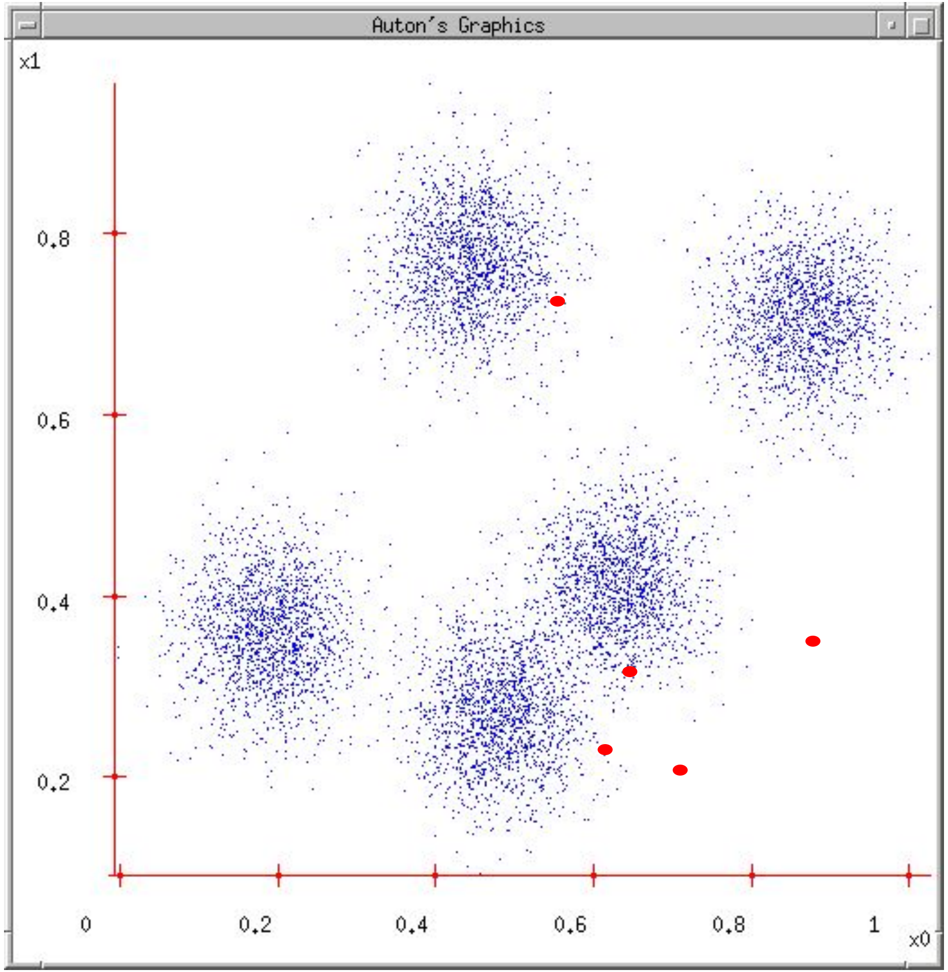
1. Ask user how many clusters they'd like. *(e.g. k=5)*



K-means

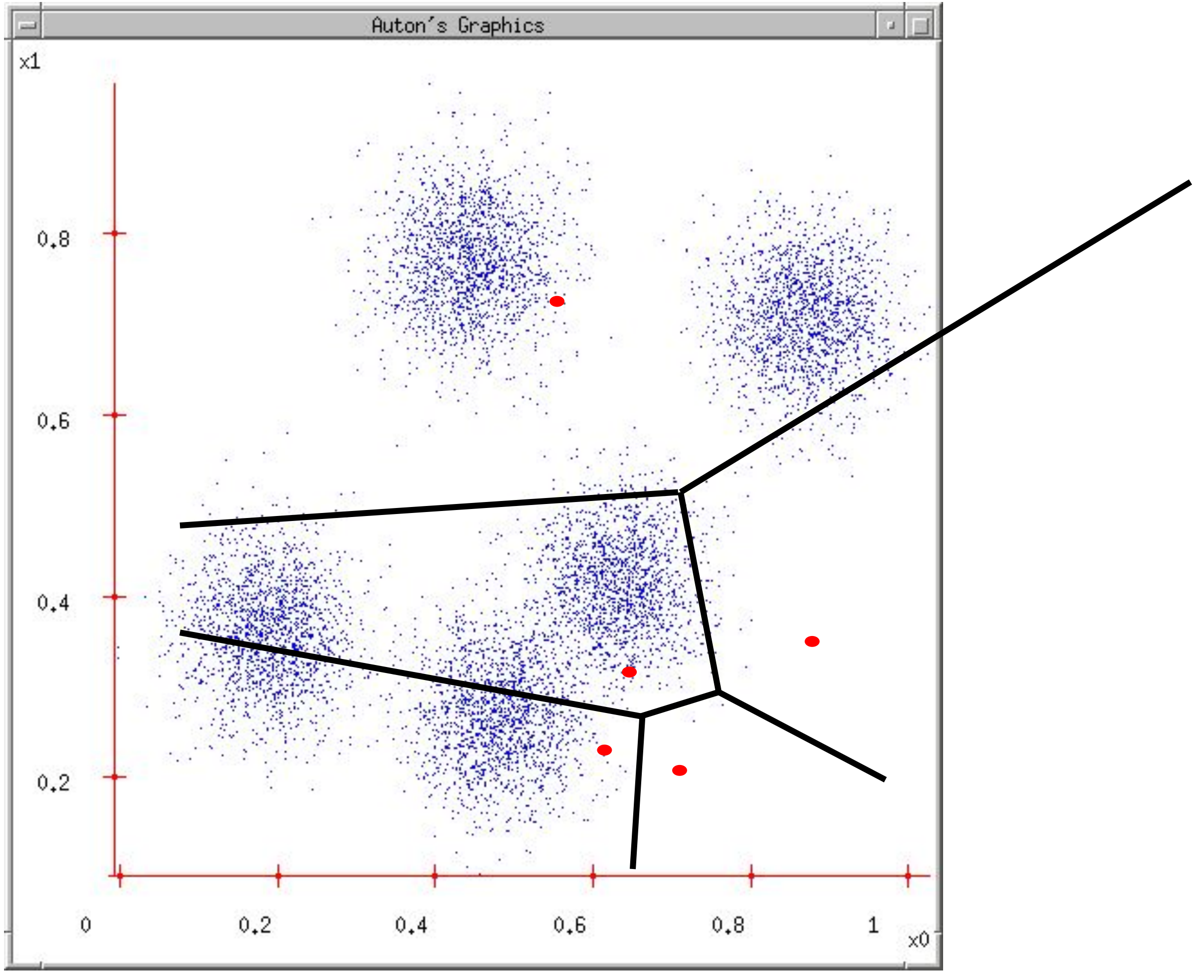
1. Ask user how many clusters they'd like. *(e.g. k=5)*

2. Randomly guess k cluster Center locations



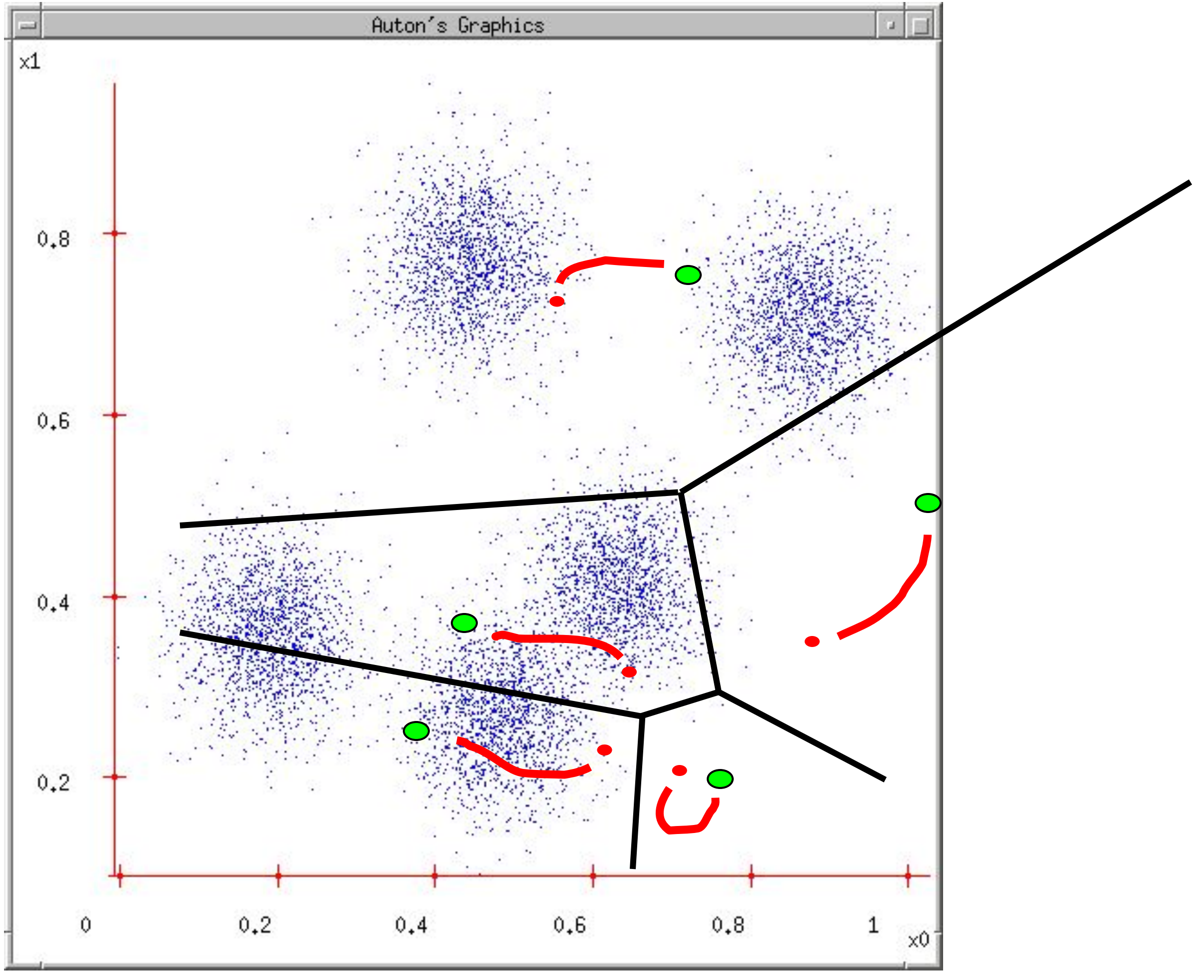
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



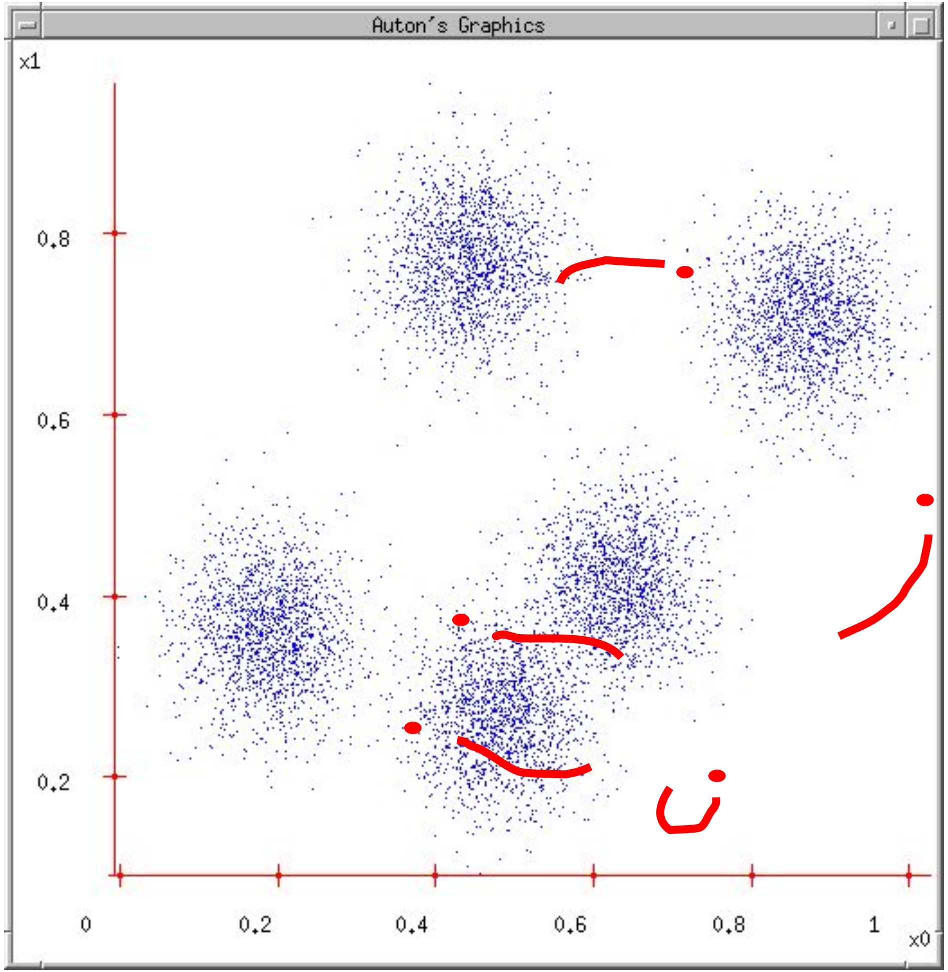
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!



K-Means is guaranteed to converge

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

Recomputation decreases average distance

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

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

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

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

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

SAPIENZA
UNIVERSITÀ DI ROMA

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 != 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.

Initialization of centroids

- 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

Complexity of K-Means

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

SAPIENZA UNIVERSITÀ DI ROMA

Evaluation

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 Evaluation

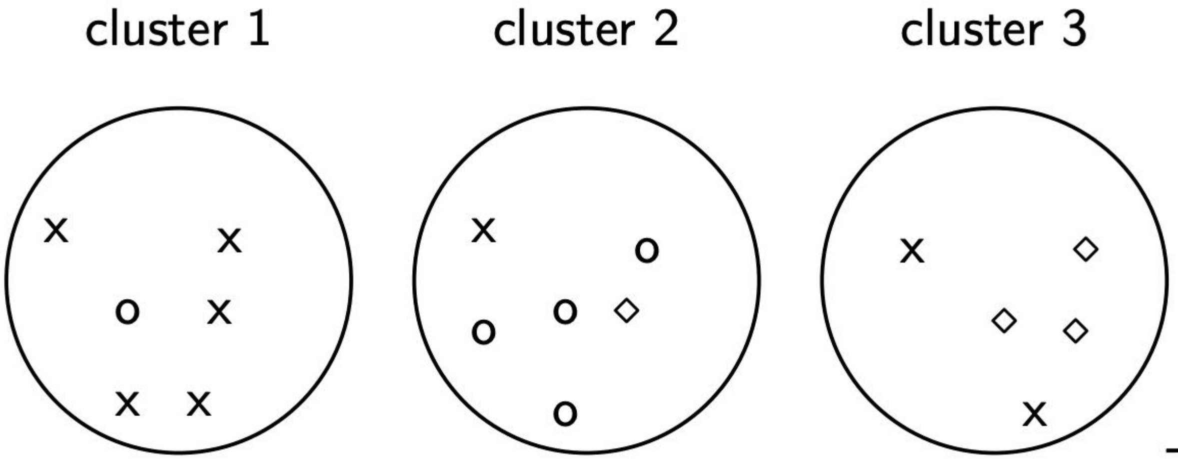
- Based on a gold standard data set, e.g., the Reuters collection we also used for the evaluation of classification
- Goal: Clustering should reproduce the classes in the gold standard
- (But we only want to reproduce how documents are divided into groups, not the class labels.)
- First measure for how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

External Criteria: Purity

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

SAPIENZA
UNIVERSITÀ DI ROMA

Computing Purity: An Example



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

SAPIENZA
UNIVERSITÀ DI ROMA

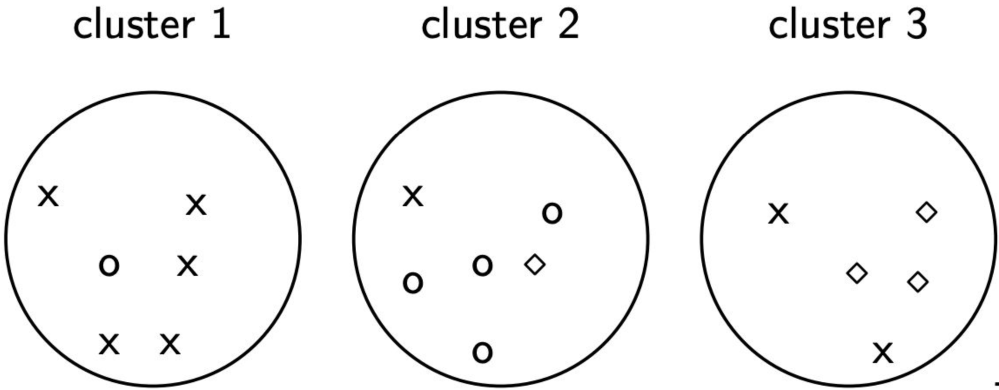
External Criteria: Rand

- Purity can be increased easily by increasing K – a measure that does not have this problem: Rand index.
- Definition: $\mathrm{RI} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}$
- 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.
- TP+FN+FP+TN = chose(N, 2) for N documents
-
- Each pair is either positive or negative (the clustering puts the two documents in the same or in different clusters) . . .
    - . . . and either “true” (correct) or “false” (incorrect): the clustering decision is correct or incorrect

Rand Index: Example



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:

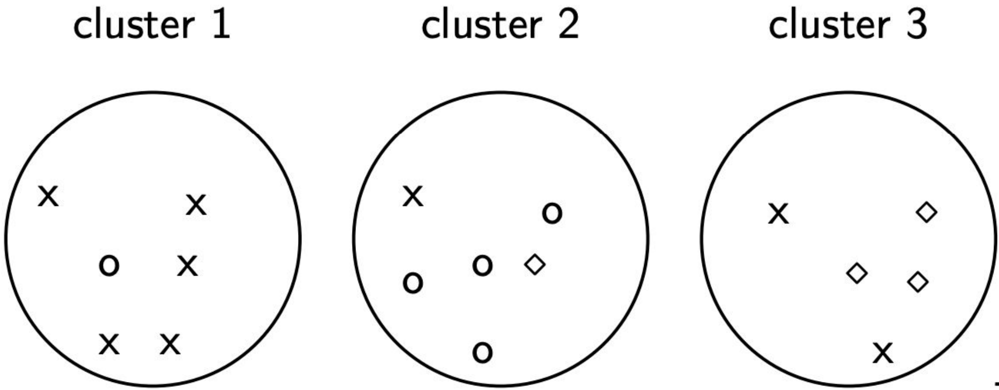
$$\mathrm{TP}+\mathrm{FP}=\binom{6}{2}+\binom{6}{2}+\binom{5}{2}=40$$

Of these, the x pairs in cluster 1, the o pairs in cluster 2, the ◊ 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, FP = 40 − 20 = 20. FN and TN are computed similarly

Rand Index: Example



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

RI is then

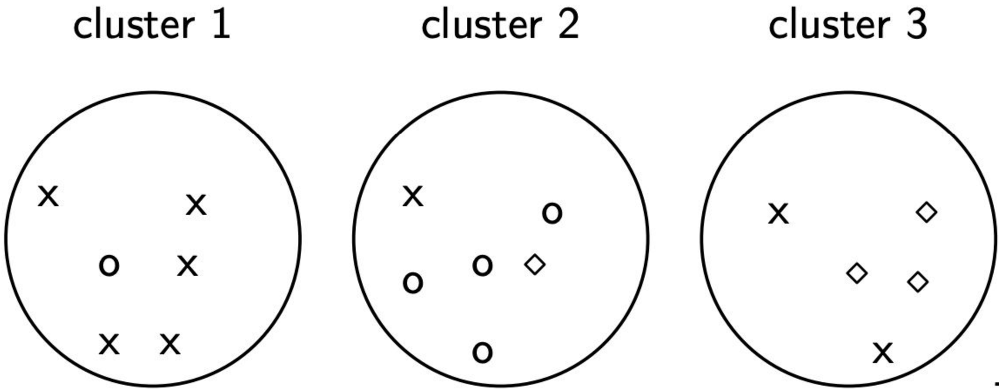
$$(20 + 72)/(20 + 20 + 24 + 72) \approx 0.68.$$

SAPIENZA UNIVERSITÀ DI ROMA

Two other external evaluation 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



	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

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 clustering of results 10 clusters are enough.
- 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.

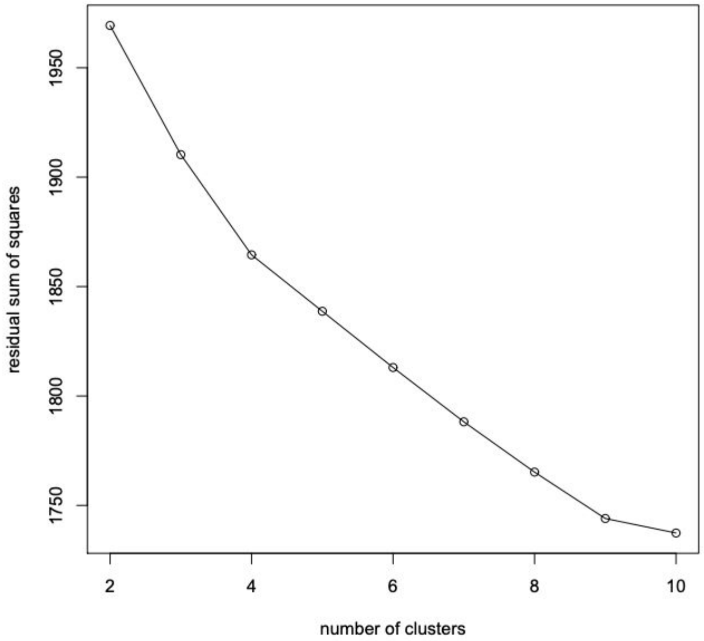
How many clusters?

- 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

How many clusters?

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

Finding the knee on the curve



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

Takeaway Messages

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