Information Storage and Retrieval

CSCE 670
Texas A&M University
Department of Computer Science & Engineering
Instructor: Prof. James Caverlee

Clustering
1 March 2017

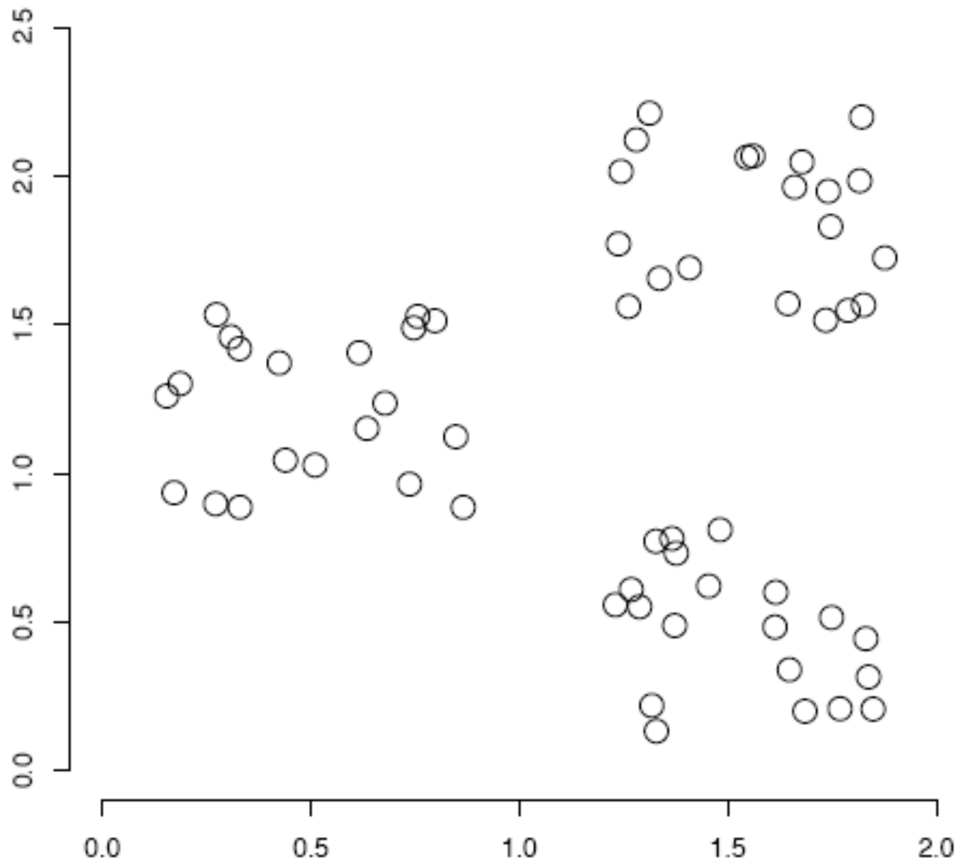
Today

- Introduction
- Clustering in IR
- K-means
- How many clusters? K?

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



- **How would you design an algorithm for finding these three clusters?**

Clustering vs. Classification

- Clustering: **unsupervised** learning
- Classification: **supervised** learning
- Classification: Classes are human-defined and 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

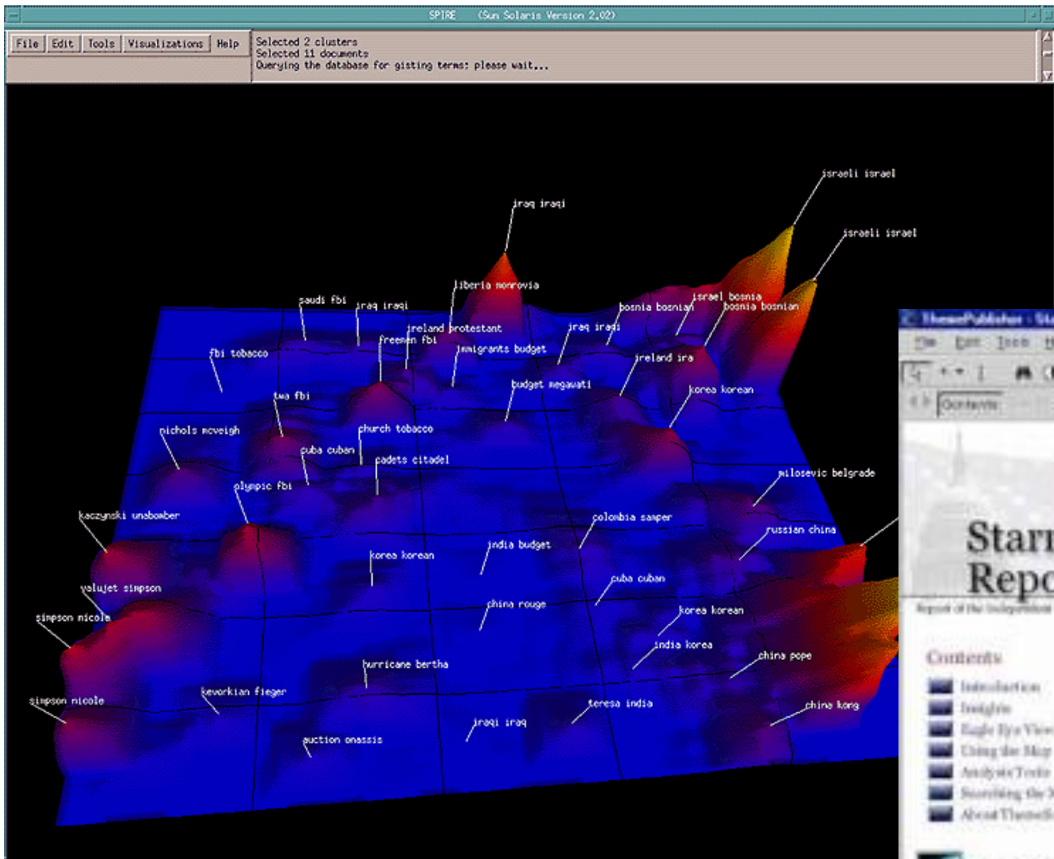
Result set clustering for better navigation

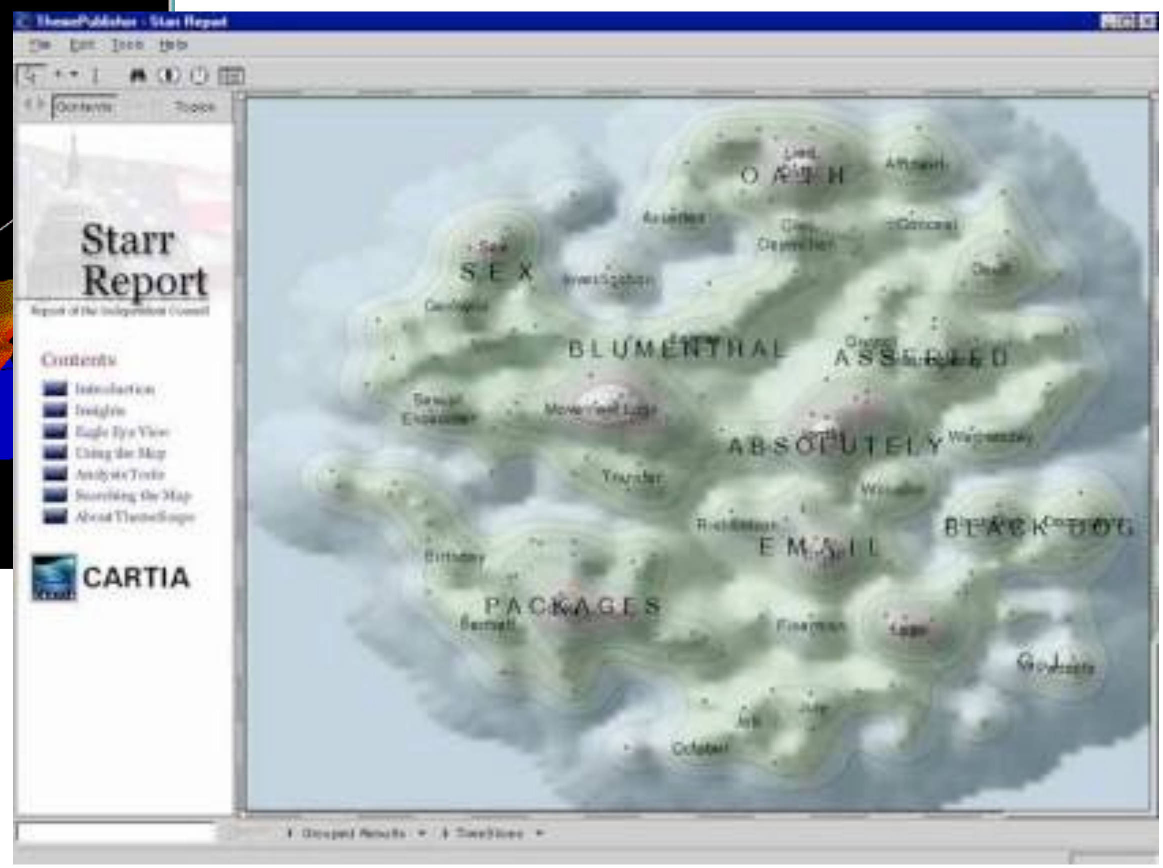
Clusty

Global clustering for improved navigation

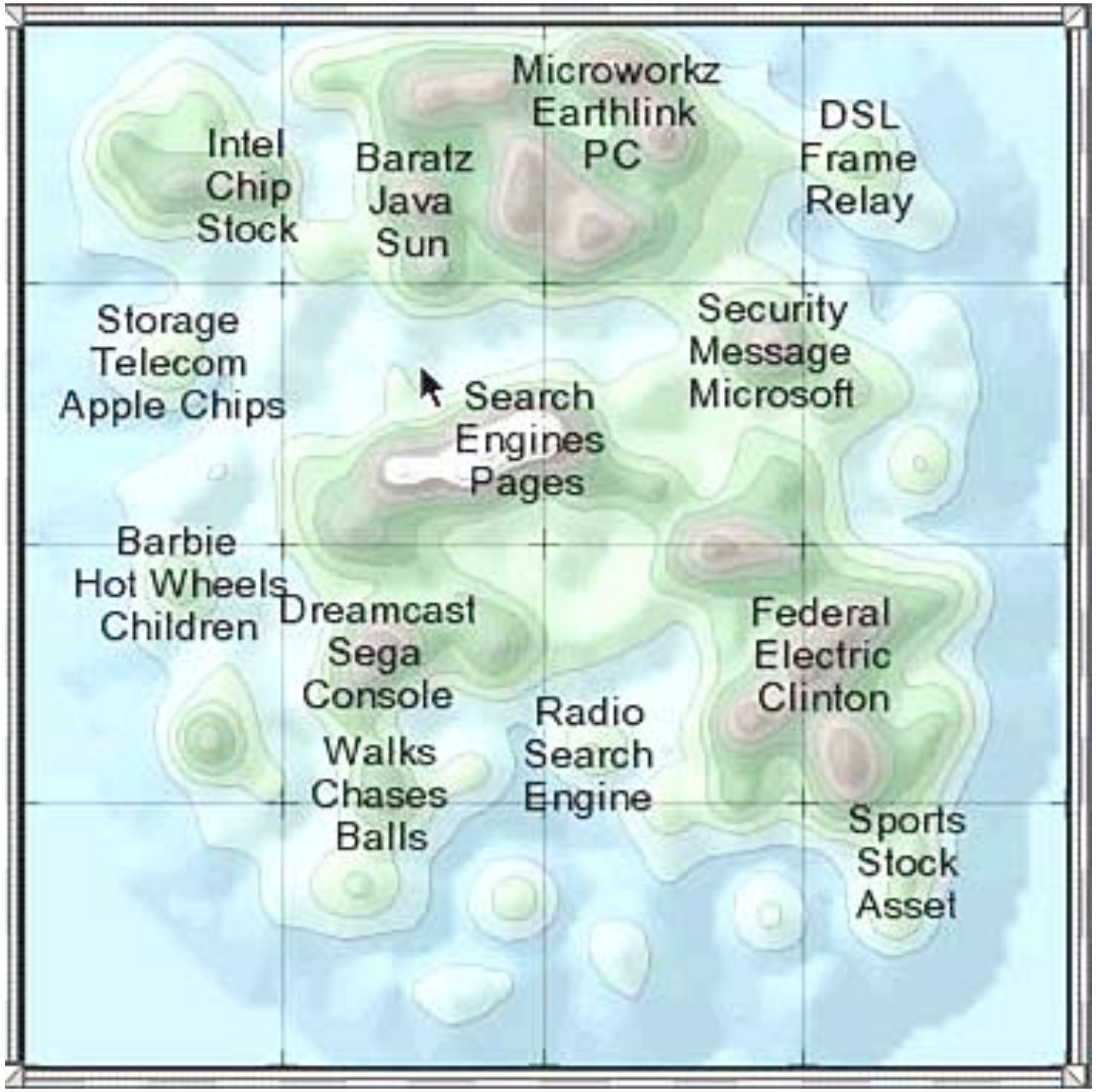
Google news

Visualizing a document collection






Bse
Meat
Beef
Meat
Animals
Cost
Fraud
Olive Oil
Production
Mobile
Service
Telephone
Feed
Bovine
Animals
Black
Fraud
Eaggf
Health
Region
Products
Plant
Italian
Production
Customs
Origin
Fraud
Television
Advertising
Broadcasting
Kaleidoscope
Costs
Ecu
Tobacco
Cigarette Smuggling
Products
Universal Service
Telecommunications
Service
Projects
Misuse
Reading
Cultural
Books
Books Reading
Scientific
Experts
Health
Restore
Committee
Irregularities
Telematics
Transport
Projects
Education
Training
Mobility


Microworkz
Earthlink
PC
Intel
Chip
Stock
Baratz
Java
Sun
DSL
Frame
Relay
Storage
Telecom
Apple Chips
Security
Message
Microsoft
Search
Engines
Pages
Barbie
Hot Wheels
Children
Dreamcast
Sega
Console
Federal
Electric
Clinton
Radio
Search
Engine
Walks
Chases
Balls
Sports
Stock
Asset

Clustering for improving recall

- **Cluster hypothesis.** Documents in the same cluster behave similarly with respect to relevance to information needs.
- Therefore, to improve search recall
  - Cluster docs in corpus 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 for clustering

- Representation for clustering
  - **Document representation**
    - Vector space? Normalization?
  - Need a notion of **similarity/distance**
- **How many clusters?**
  - Fixed a priori?
  - Completely data driven?
    - Avoid "trivial" clusters - too large or small
      - In an application, if a cluster's too large, then for navigation purposes you've wasted an extra user click without whittling down the set of documents much.

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

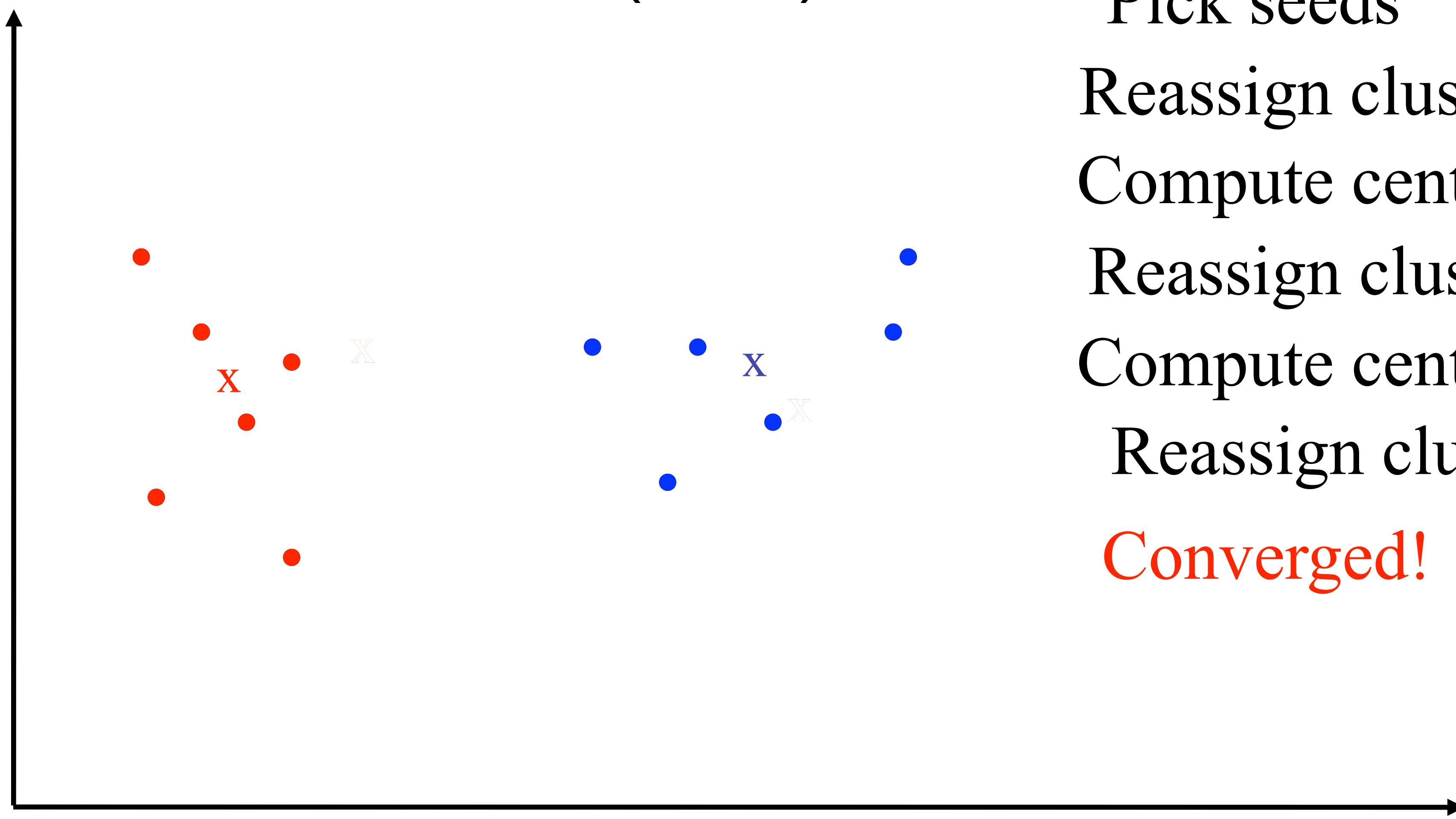
- Hard clustering: Each document belongs to exactly one cluster.
  - More common and easier to do
- Soft clustering: A document can belong to more than one cluster.
  - Makes more sense for applications like creating browsable hierarchies
- You may want to put a pair of sneakers in two clusters: (i) sports apparel and (ii) shoes
- You can only do that with a soft clustering approach.
- See Book Chapter 16.5

Flat algorithms

- Flat algorithms compute a partition of N documents into a set of K clusters.
- <u>**Given:**</u> a set of documents and the number K
- <u>**Find:**</u> a partition in K clusters that optimizes the chosen partitioning criterion
- Global optimization: exhaustively enumerate partitions, pick optimal one
  - Not tractable
- Effective heuristic methods: K-means and K-medoids algorithms

K-means

K Means Example
(*K*=2)



Pick seeds
Reassign clusters
Compute centroids
Reassign clusters
Compute centroids
Reassign clusters

Converged!

K-means

- Objective/partitioning criterion: minimize the average squared difference from the centroid
- Assumes documents are real-valued vectors.
- Clusters based on *centroids* (aka the *center of gravity* or mean) of points in a cluster, *w*:

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

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

$(\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_K) \leftarrow$ 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 $j \leftarrow \arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$

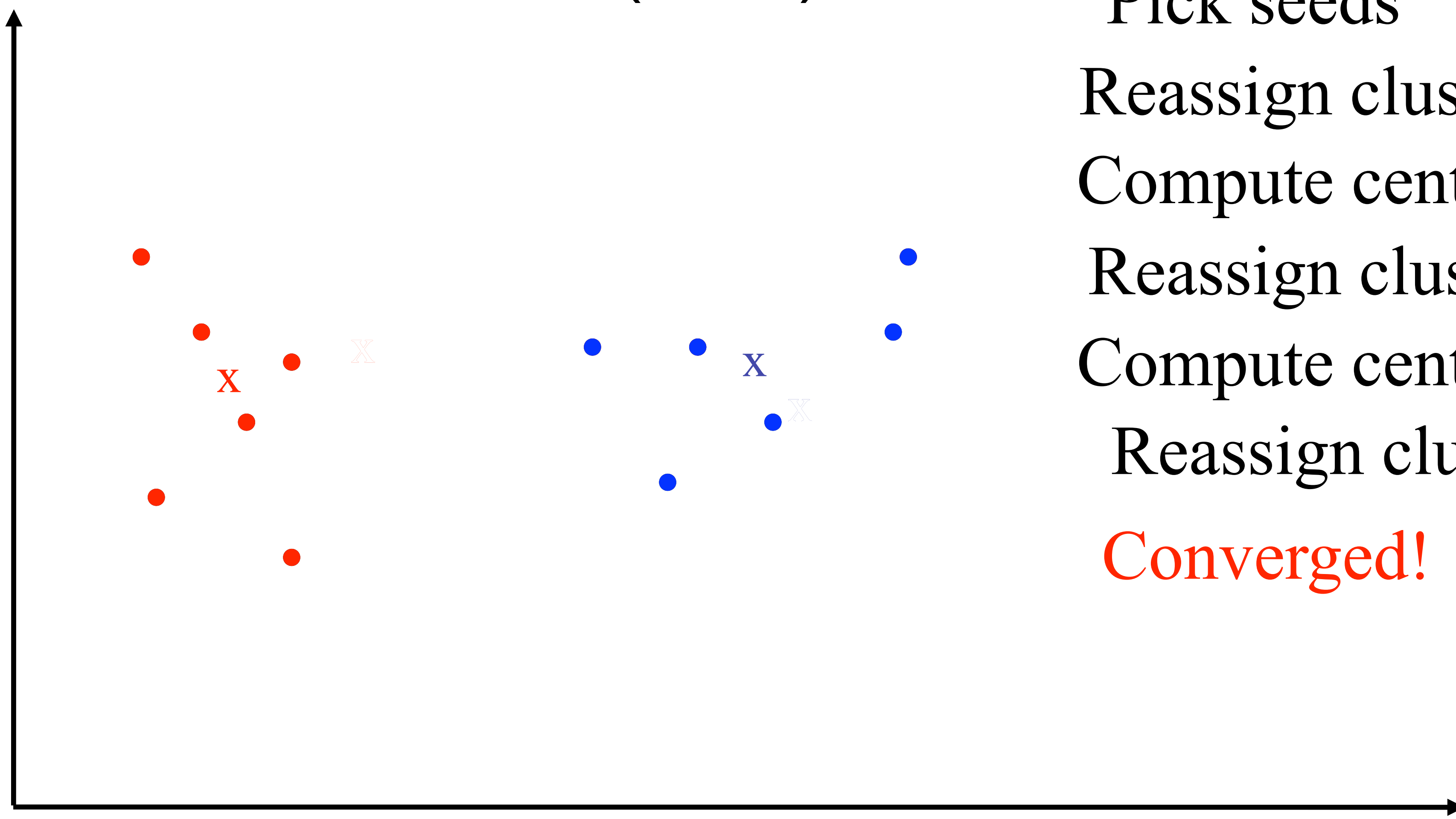
    $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ *(reassignment of vectors)*

  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 Example
(*K*=2)



Pick seeds
Reassign clusters
Compute centroids
Reassign clusters
Compute centroids
Reassign clusters

Converged!

- http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html

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).

Recomputation decreases average distance

RSS = residual sum of squares (the goodness measure G)

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

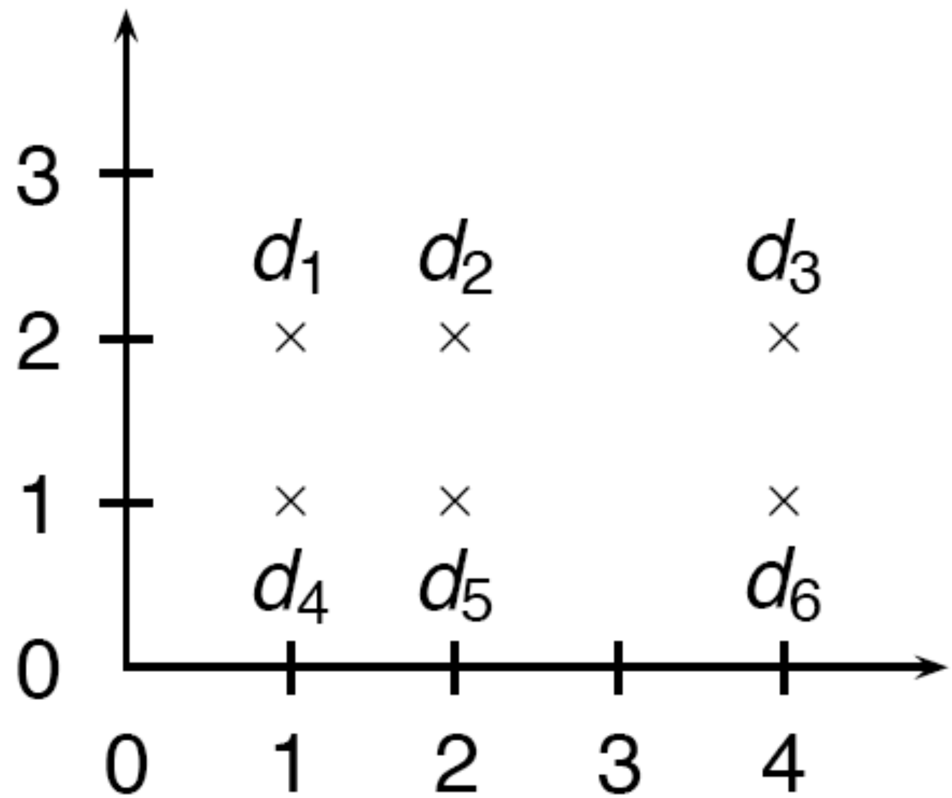
$$v_m = \frac{1}{|\omega_k|} \sum_{\vec{x}\in\omega_k} x_m$$

This is the componentwise definition of the centroid!
We minimize RSSk when the old centroid is replaced with the new centroid. RSS, the sum of the RSSk , must then also decrease during recomputation.

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 of suboptimal clustering!!!! LOL



- What is the optimal clustering for K=2?
- What happens when our seeds are: d2, d5?

Initialization of K-means

- Results can vary based on random seed selection.
- Some seeds can result in poor convergence rate, or convergence to sub-optimal clusterings.
  - Select good seeds using a heuristic (e.g., doc least similar to any existing mean)
  - Try out multiple starting points
  - Initialize with the results of another method.

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.

How many clusters?

Hmm...

- **<u>Either: Number of clusters K is given</u>**.
  - Then partition into K clusters
  - K might be given because there is some external constraint. Example: You cannot show more than 10–20 clusters on a screen.
- **<u>Or: Finding the "right" number of clusters is part of the problem.</u>**
  - Given docs, find K for which an optimum is reached.
  - How to define "optimum"?
  - Why can't we use RSS or average distance from centroid?

Simple objective function for K

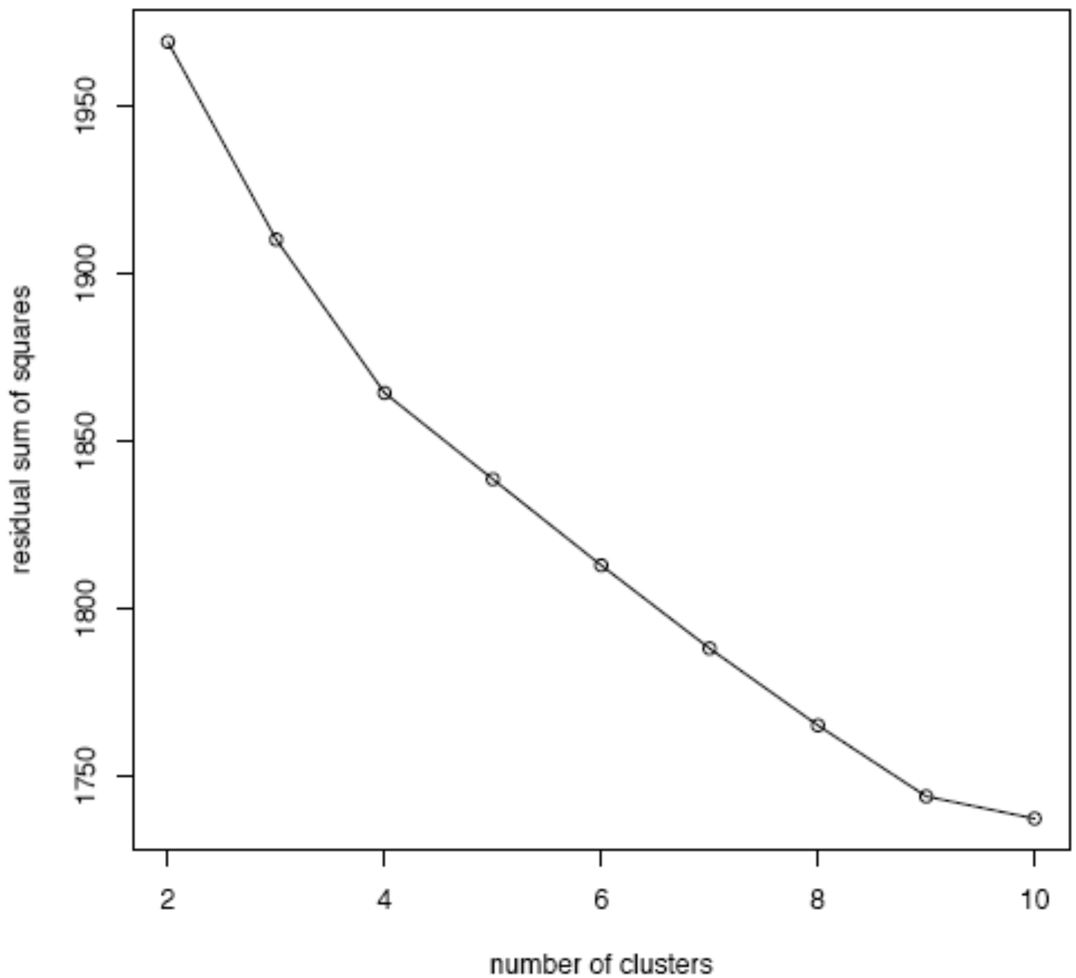
- Basic idea:
  - Start with 1 cluster (K = 1)
  - Keep adding clusters (= keep increasing K)
  - Add a penalty for each new cluster
- Trade off cluster penalties against average squared distance from centroid
- Choose K with best tradeoff

Simple objective function for K

- Given a clustering, define the cost for a document as (squared) distance to centroid
- Define total distortion RSS 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: $RSS + K\lambda$
- Select K that minimizes $(RSS + K\lambda)$
- Still need to determine good value for λ . . .

What if $\lambda = 0$?

Finding the “knee” in the curve



- Pick k where curve flattens (4, 9)