Information Retrieval & Social Web

CS 525/DS 595

Worcester Polytechnic Institute
Department of Computer Science
Instructor: Prof. Kyumin Lee

Previous Class…

Classification

Previous Class…

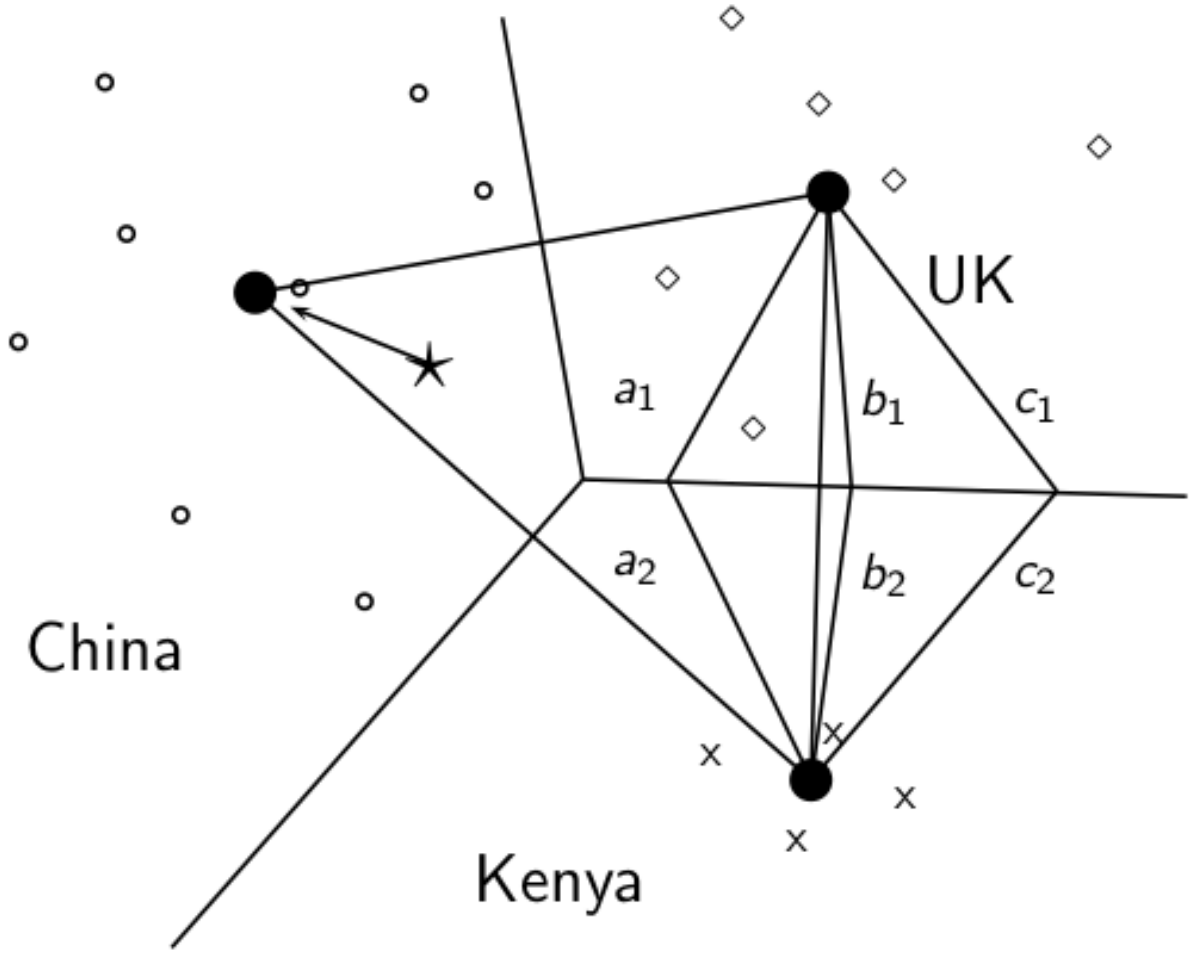
Classification

Vector Space Classification

Previous Class…

Rocchio

Rocchio illustrated : a1 = a2, b1 = b2, c1 = c2

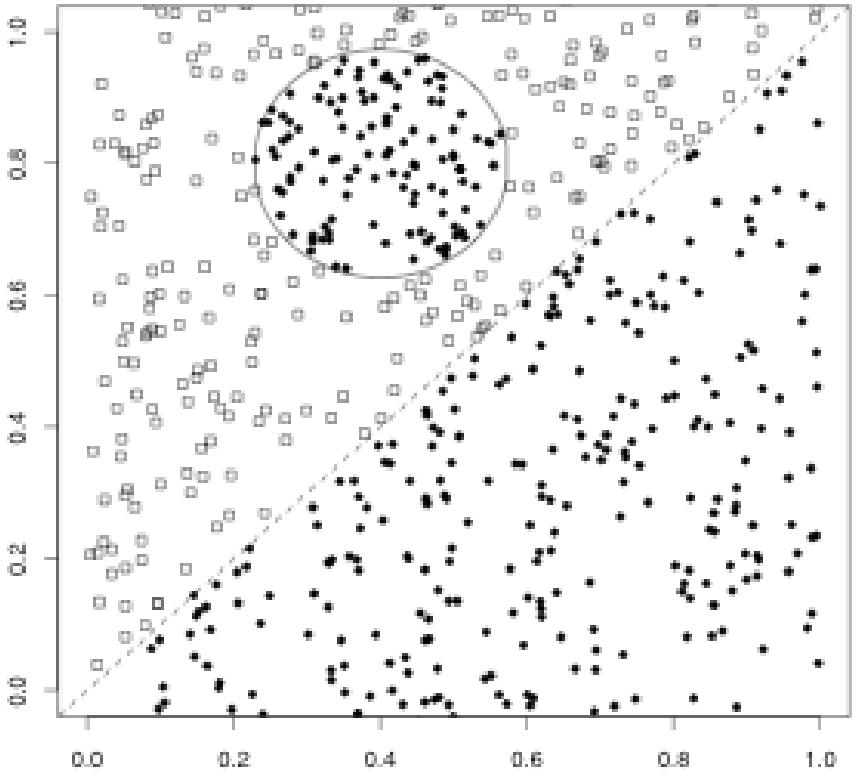


Previous Class…

Rocchio

kNN

A nonlinear problem (Rocchio vs kNN)



- Linear classifier like Rocchio does badly on this task.
- kNN will do well (assuming enough training data)

Previous Class…

Naive Bayes Classifier

The Naive Bayes Classifier

- The Naive Bayes classifier is a probabilistic classifier
- We compute the probability of a document *d* being in a class *c* as follows:

Prior

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

Posterior

- $P(c)$ is the prior probability of c.
- n_d is the length of the document. (number of tokens)
- $P(t_k|c)$ is the conditional probability of term t_k occurring in a document of class c
- $P(t_k|c)$ as a measure of how much evidence t_k contributes that c is the correct class.
- If a document's terms do not provide clear evidence for one class vs. another, we choose the c with highest $P(c)$ probability.

Previous Class…

Evaluating a Classifier ➔ precision, recall, F-meaure and accuracy

Previous Class…

What is Clustering?
➔ Unsupervised learning

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 learning = learning from raw data, as opposed to supervised data where a classification of examples is given
- A common and important task that finds many applications in IR and other places

Clustering in IR

Flat vs. Hierarchical Clustering

- Flat algorithms
  - Usually start with a random (partial) partitioning
  - Refine it iteratively
    - Main algorithm: K-means
- Hierarchical algorithms
  - Create a hierarchy
  - Bottom-up, agglomerative
  - Top-down, divisive

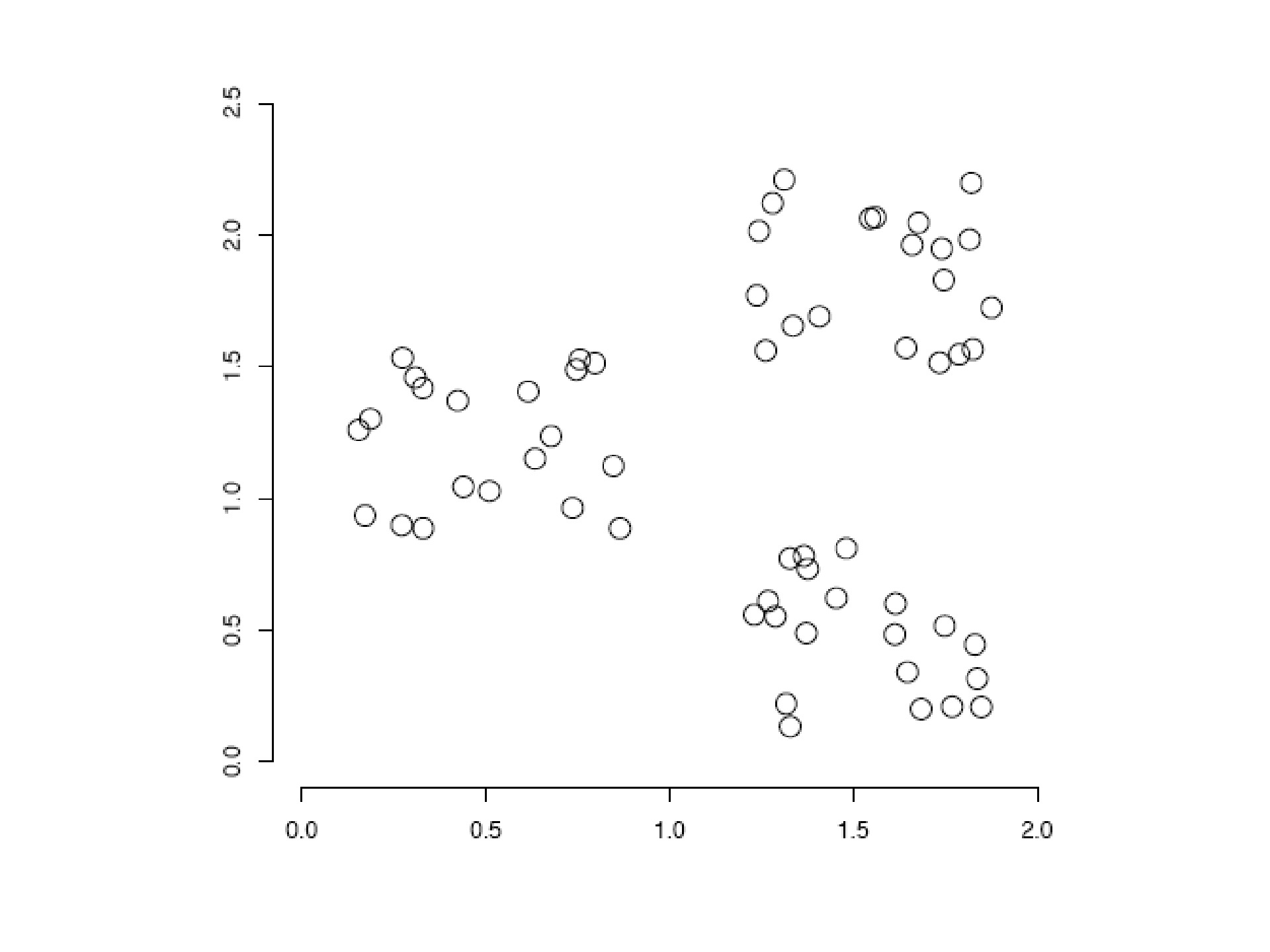
Hard vs. soft clustering

- 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.
- 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


0.0
0.5
1.0
1.5
2.0
2.5
0.0
0.5
1.0
1.5
2.0

K-means (in one slide!)

Input is **k** (the number of clusters), **data points** in Euclidean space

0. Initialize clusters by picking one point per cluster

Loop:

1. Place each point in the cluster whose current centroid is nearest
2. Find the new centroid for each cluster

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, ω:

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

- 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

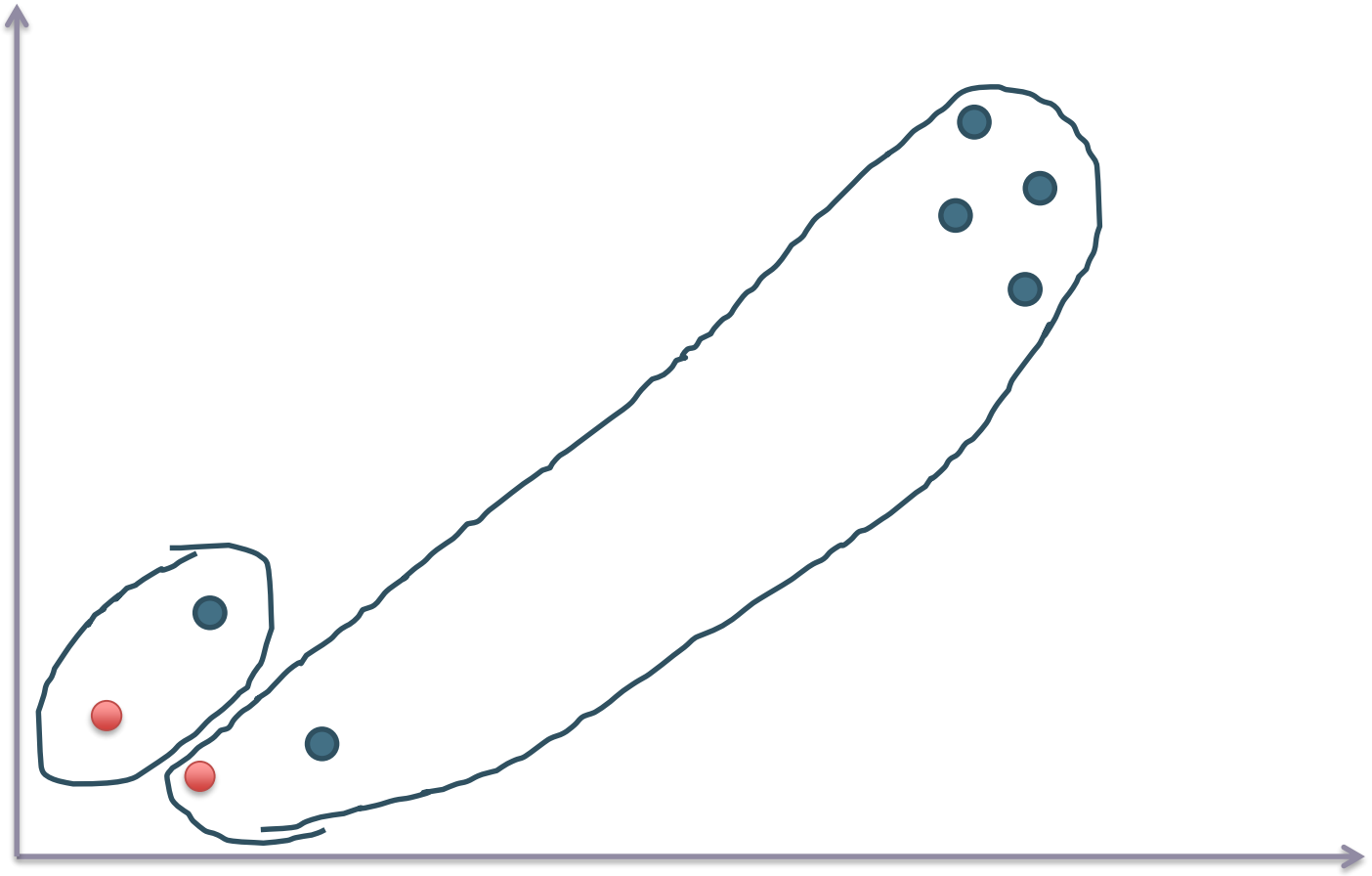
K-MEANS$(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$

1 $(\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_K) \leftarrow$ SELECTRANDOMSEEDS$(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$
2 **for** $k \leftarrow 1$ **to** K
3 **do** $\vec{\mu}_k \leftarrow \vec{s}_k$
4 **while** stopping criterion has not been met
5 **do for** $k \leftarrow 1$ **to** K
6 **do** $\omega_k \leftarrow \{\}$
7 **for** $n \leftarrow 1$ **to** N
8 **do** $j \leftarrow \arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$
9 $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ *(reassignment of vectors)*
10 **for** $k \leftarrow 1$ **to** K
11 **do** $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$ *(recomputation of centroids)*
12 **return** $\{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$

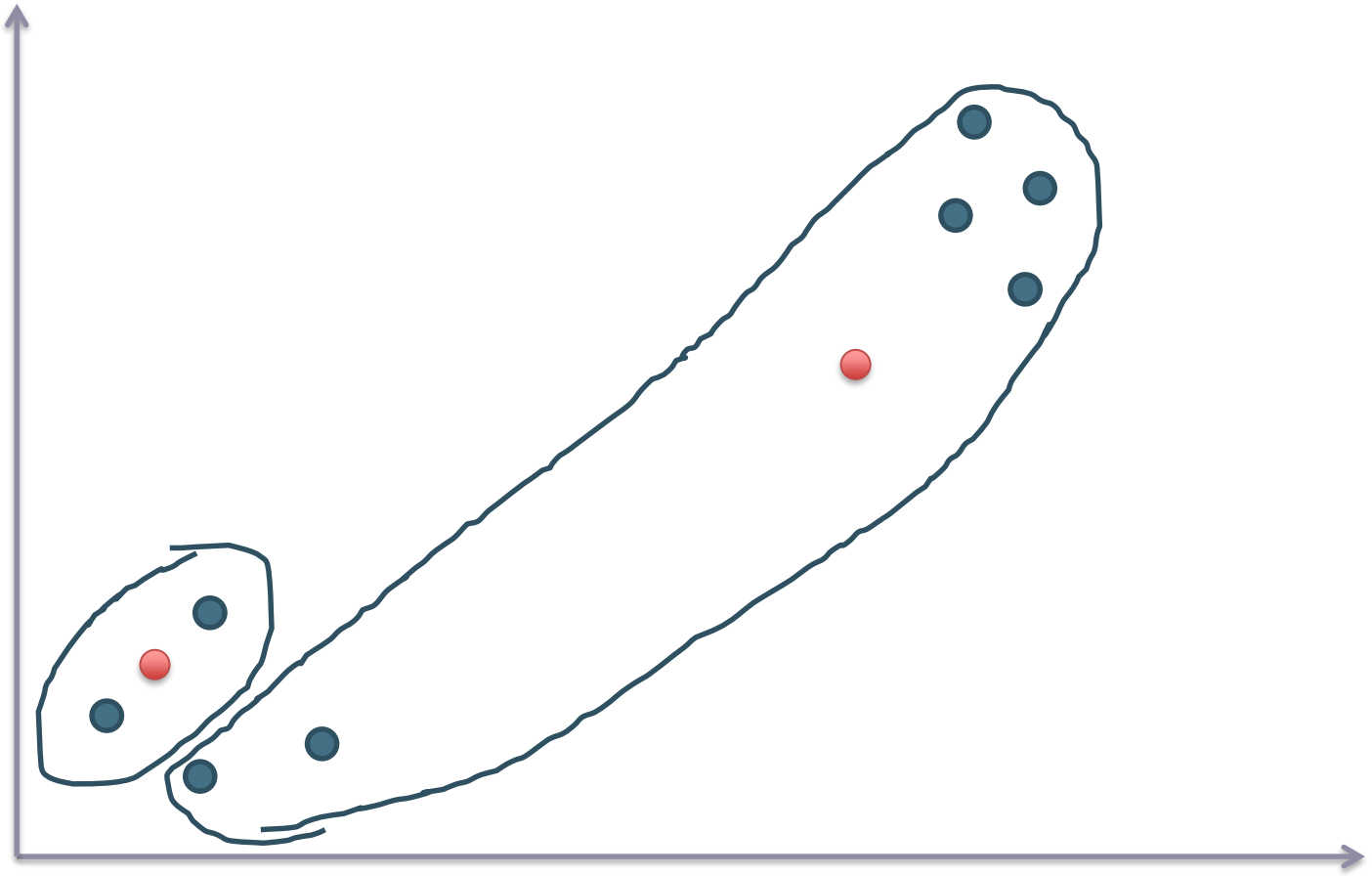
K-Means Clustering Example

K-Means Clustering Example

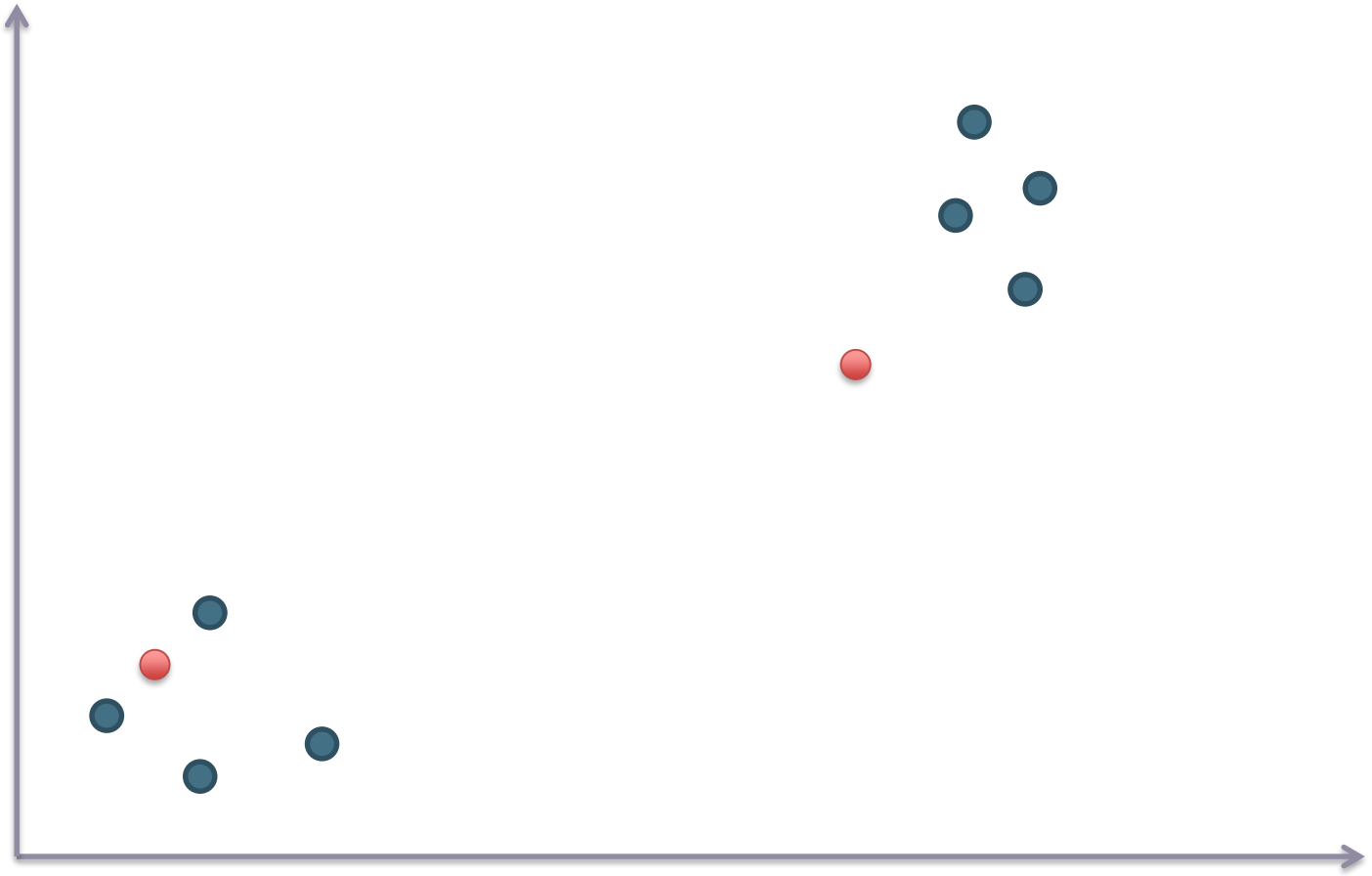
K-Means Clustering Example



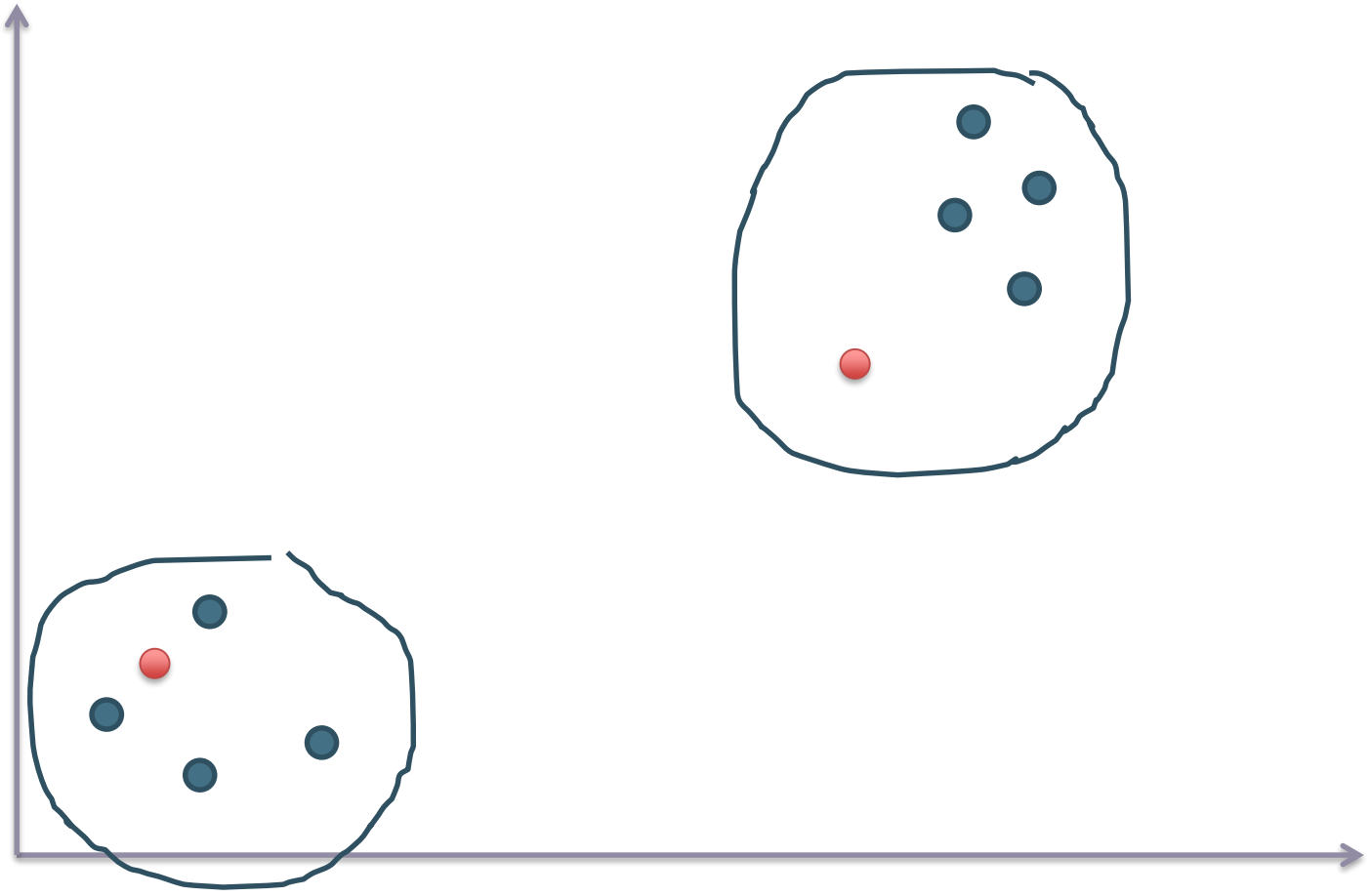
K-Means Clustering Example



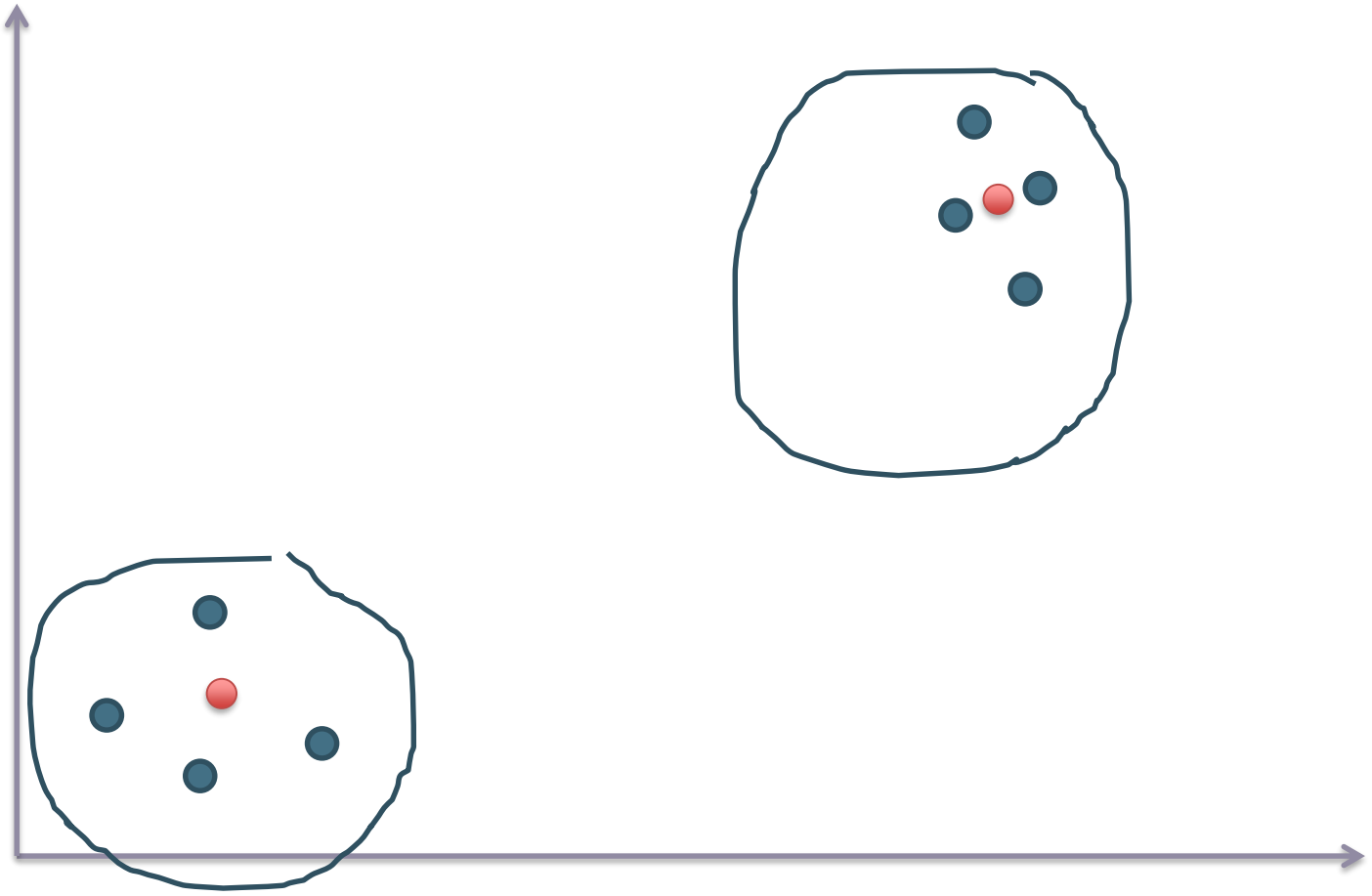
K-Means Clustering Example



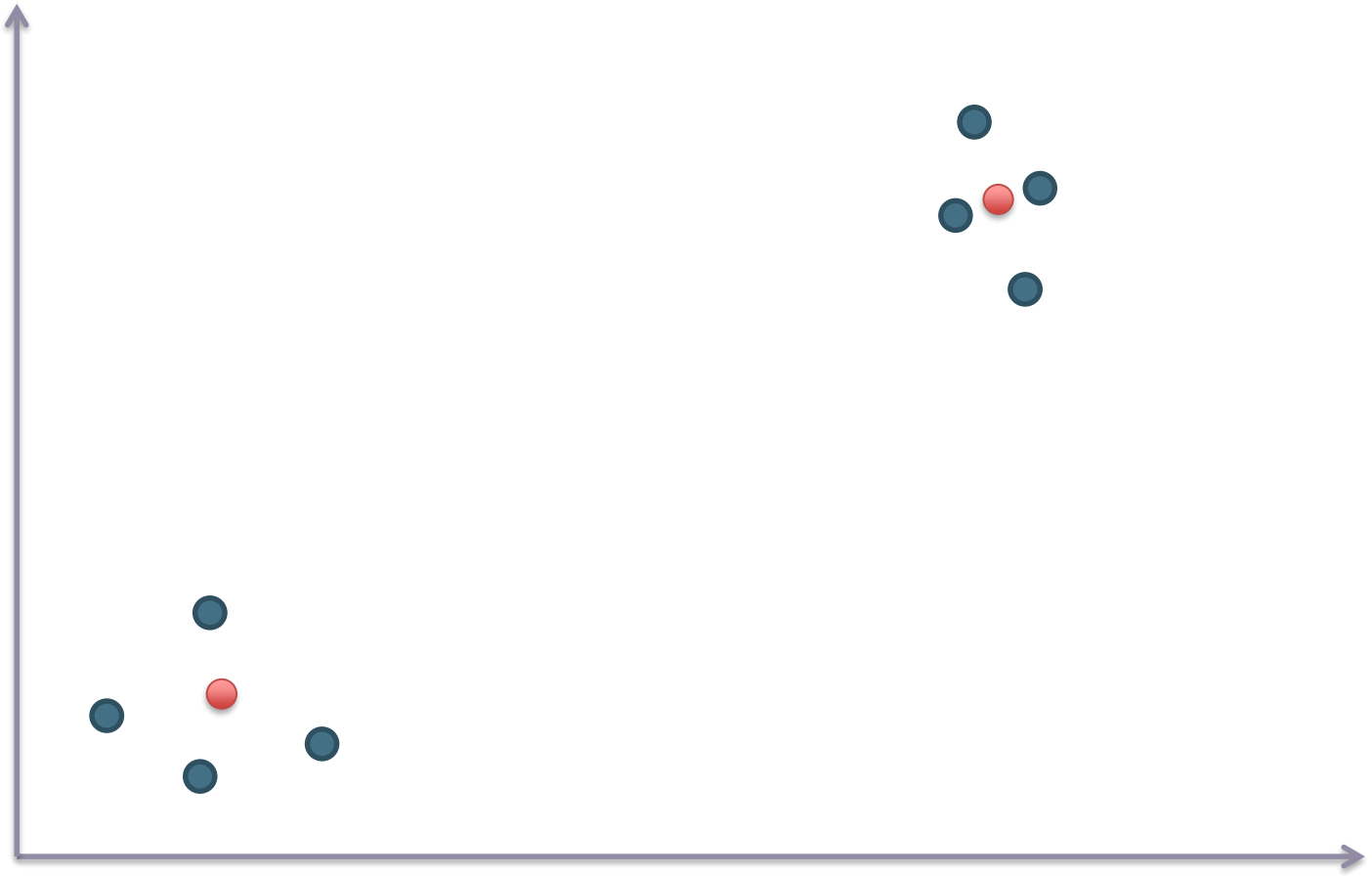
K-Means Clustering Example



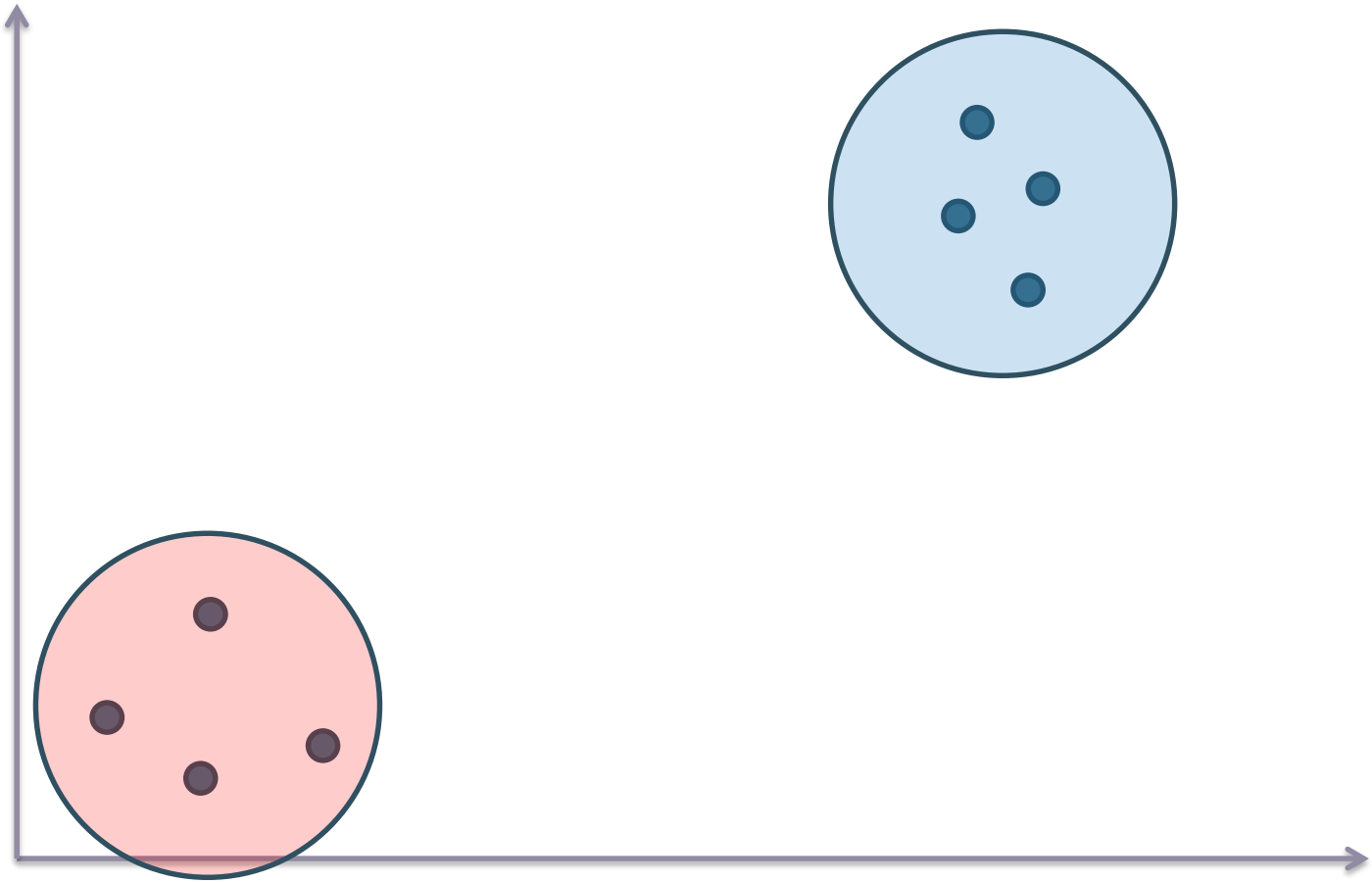
K-Means Clustering Example



K-Means Clustering Example

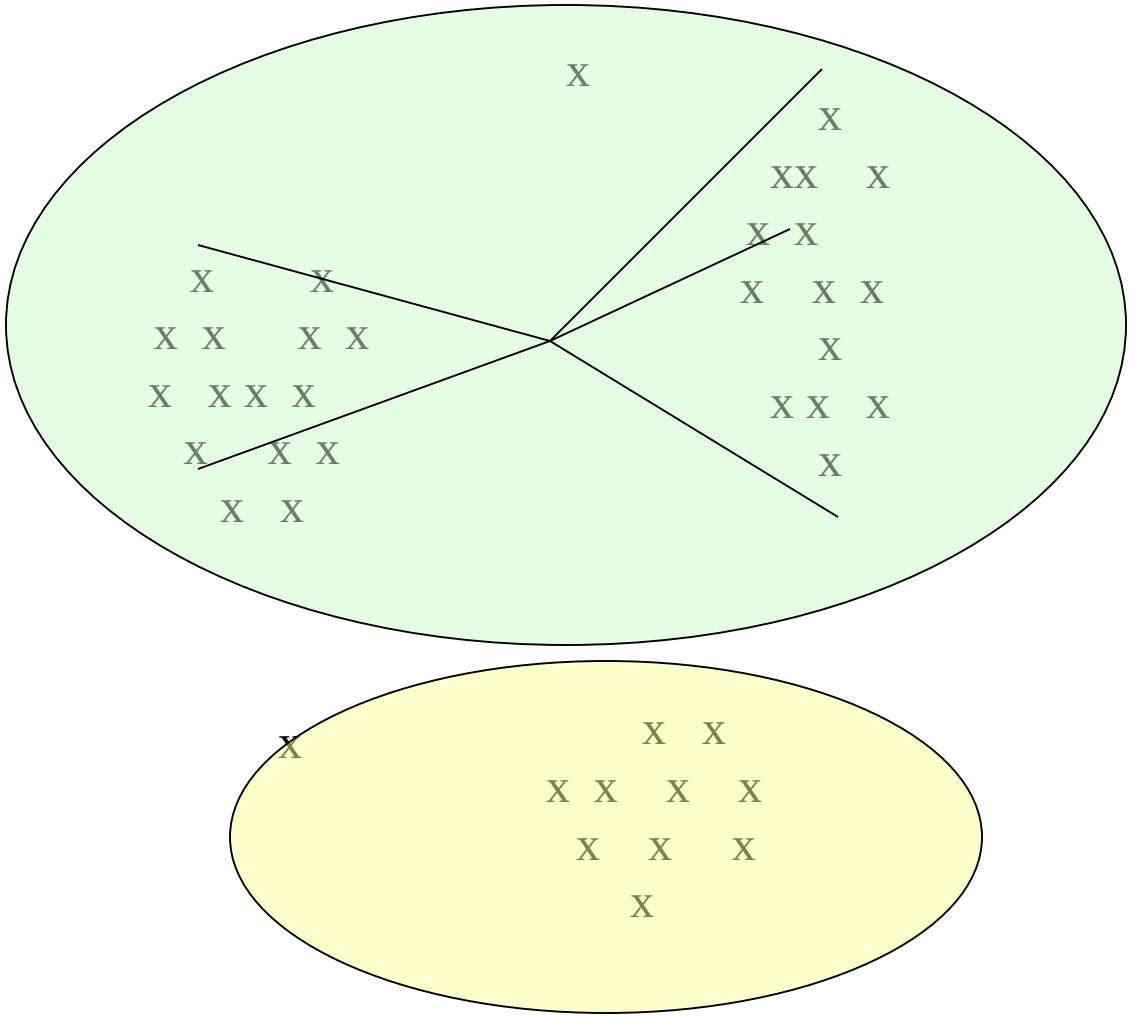


K-Means Clustering Example



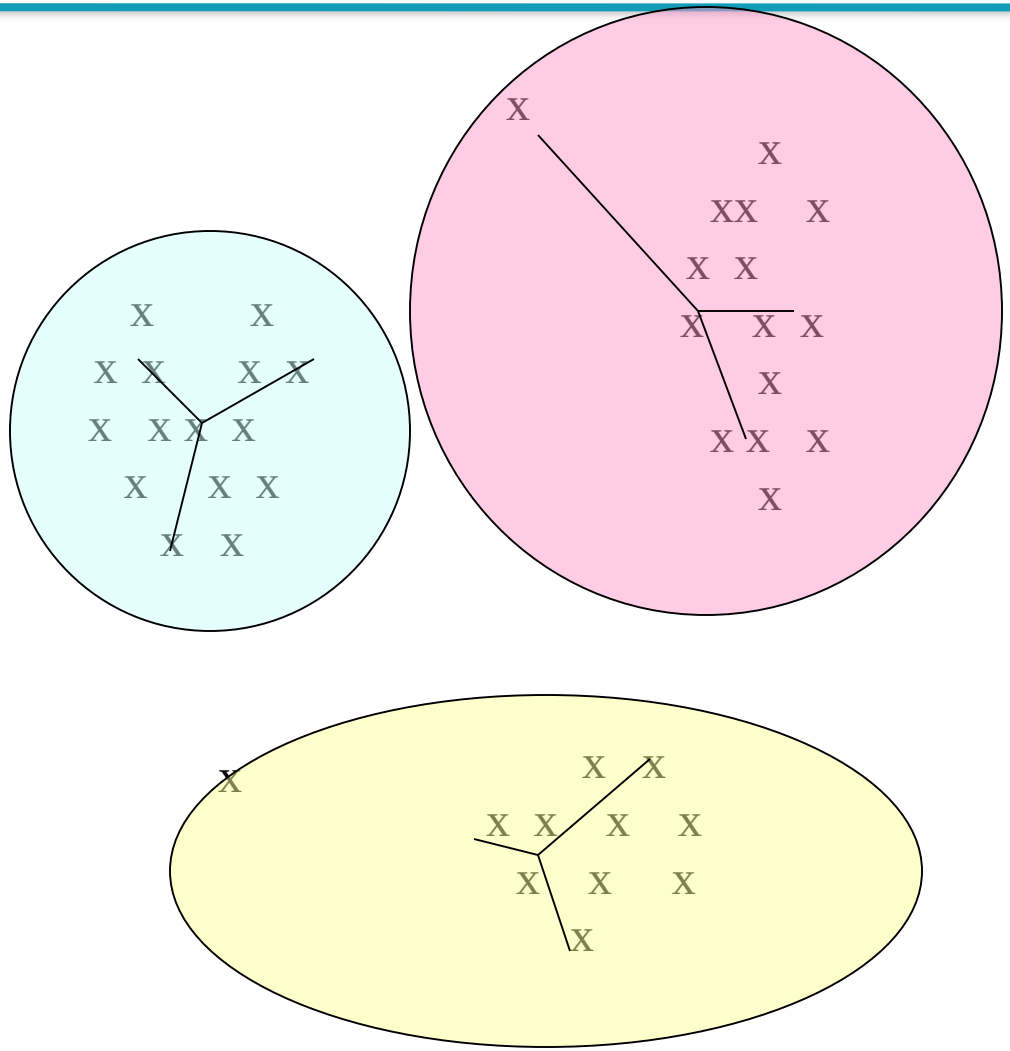
Example: Picking *k*

Too few;
many long
distances
to centroid.



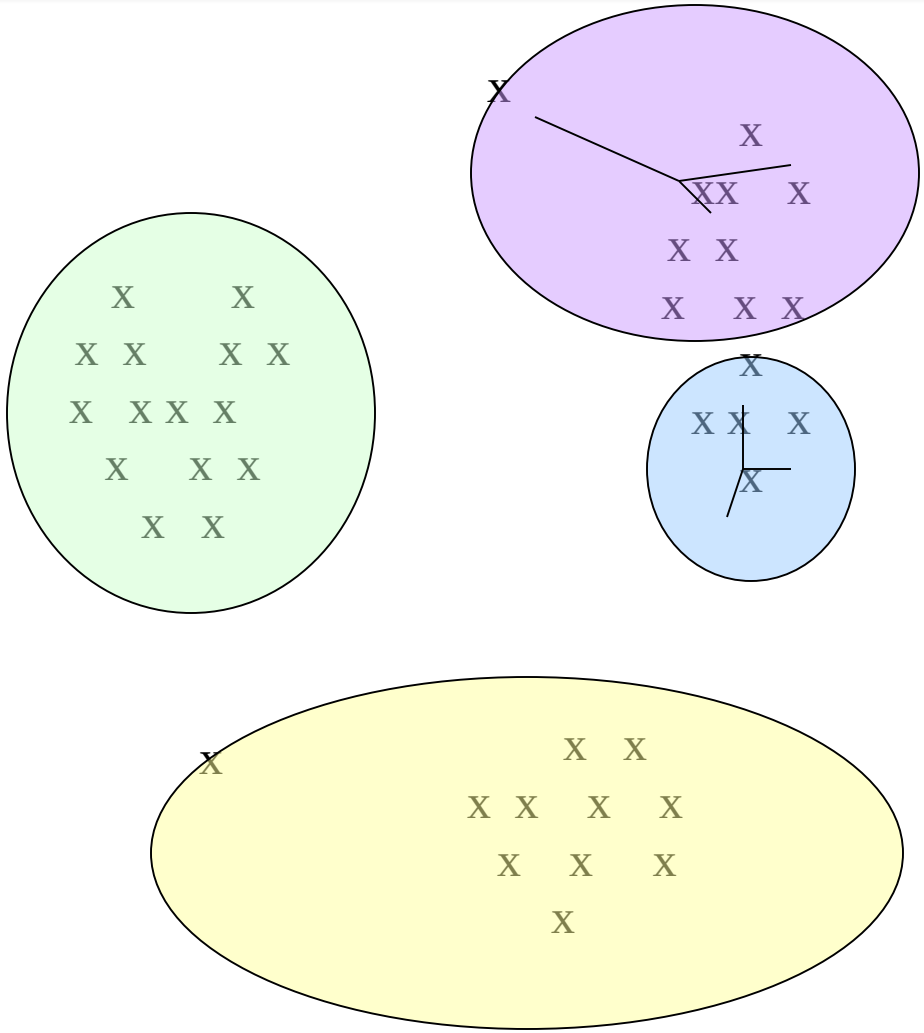
Example: Picking *k*

Just right;
distances
rather short.



Example: Picking *k*

Too many;
little improvement
in average
distance.



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.
  - There is only a finite number of clusterings
  - Thus: We must reach a fixed point.
- 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)

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

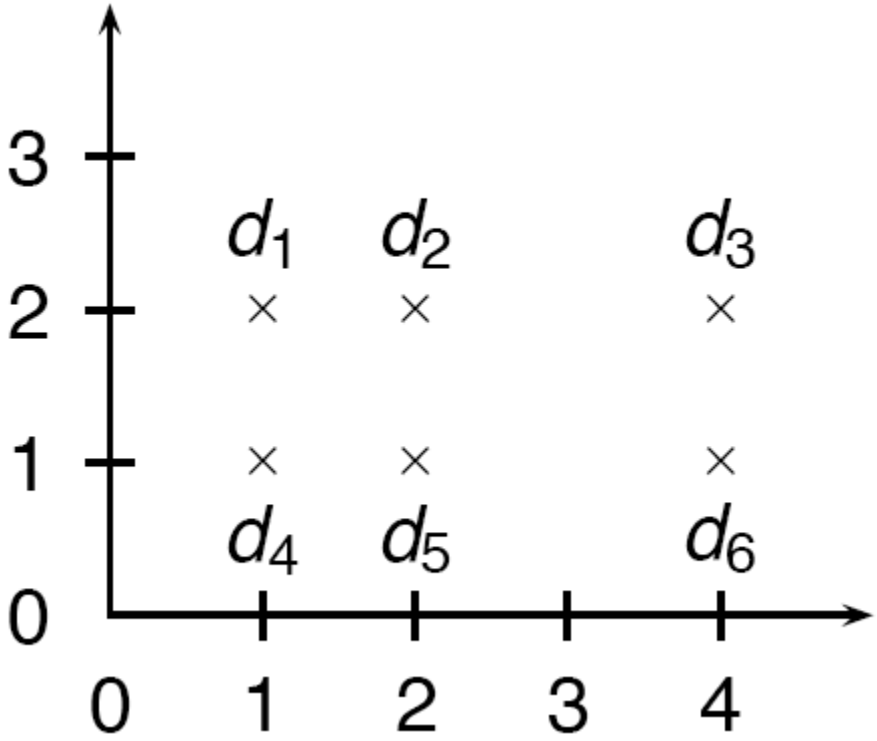
$$\mathrm{RSS} = \sum_{k=1}^{K} \mathrm{RSS}_k$$

- 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!!!!



- 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.
  - Try out multiple starting points and choose the clustering with lowest cost
  - Initialize with the results of another method such as hierarchical clustering.

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 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:
  - In the worst case, it takes superpolynomial time ($2^{\Omega(\sqrt{n})}$)
  - Refer to "How Slow is the k-Means Method?" paper.

How many clusters?

Hmm...

- **Either: Number of clusters K is given.**
  - 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.

- **Or: Finding the "right" number of clusters is part of the problem.**
  - 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?

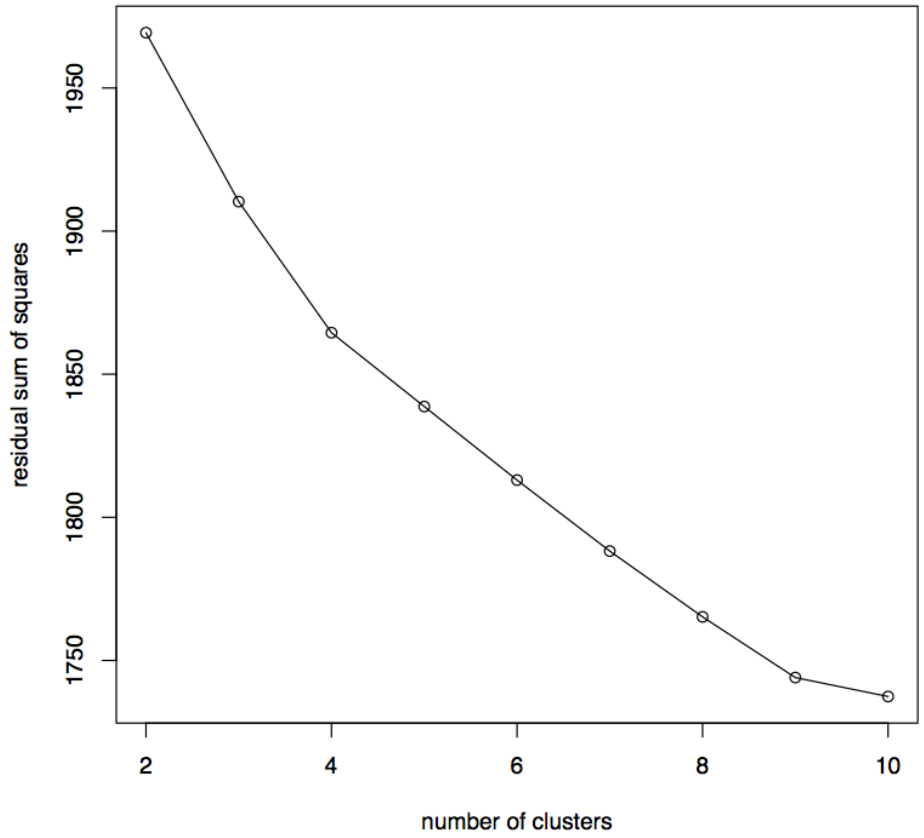
Simple objective function for *K*

- 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 the value of *K* with the 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(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*λ
- Define the total cost of a clustering as distortion plus total cluster penalty: RSS(K) + *K*λ
- Select K that minimizes (RSS(K) + *K*λ)
- Still need to determine good value for λ . . .

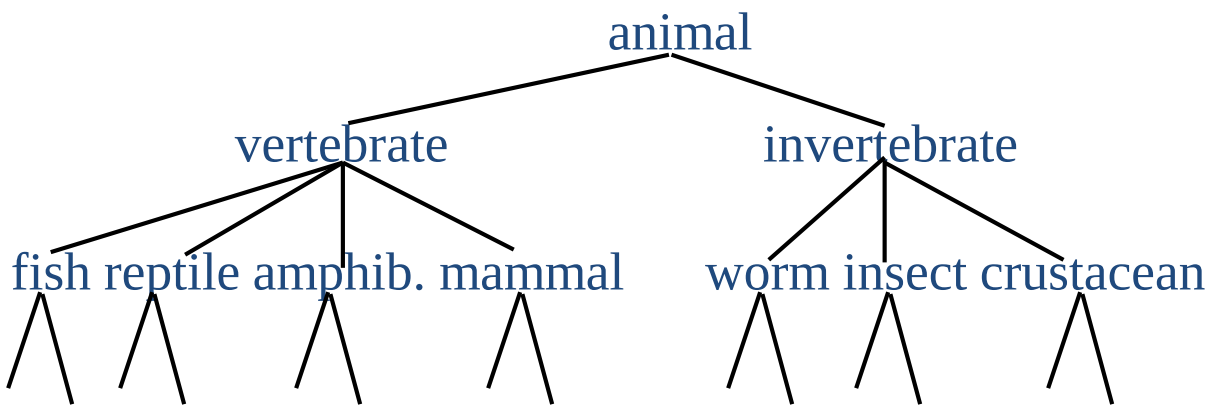
Finding the “knee” in the curve



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

Hierarchical Clustering

Hierarchical Clustering

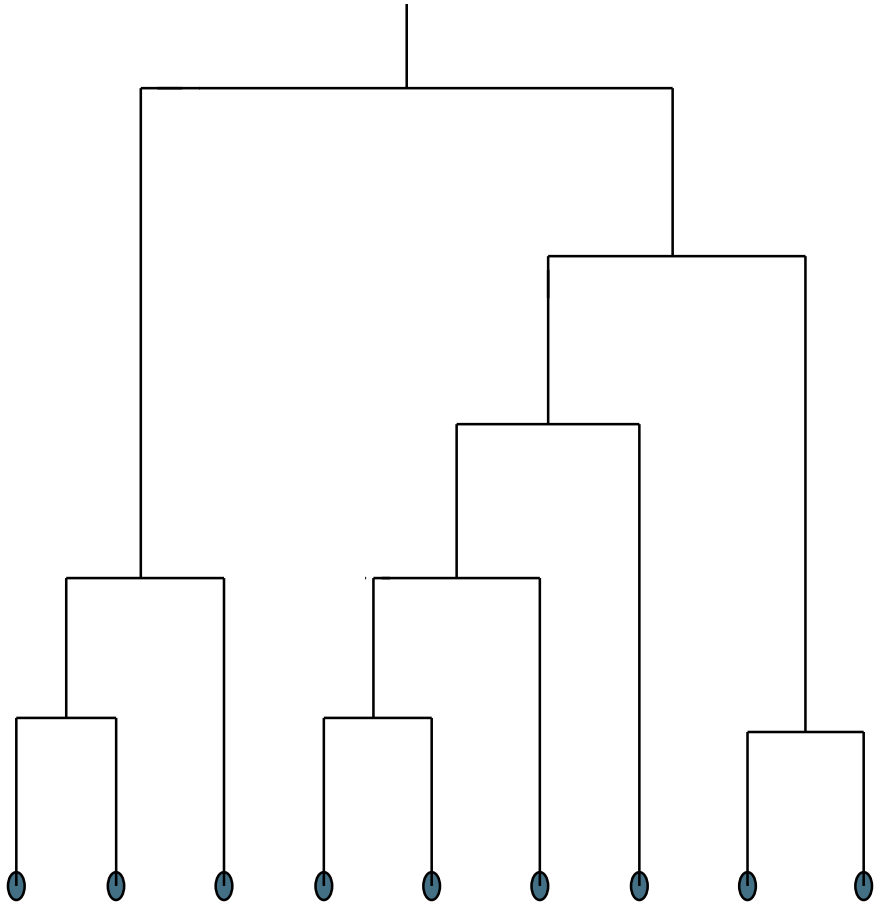


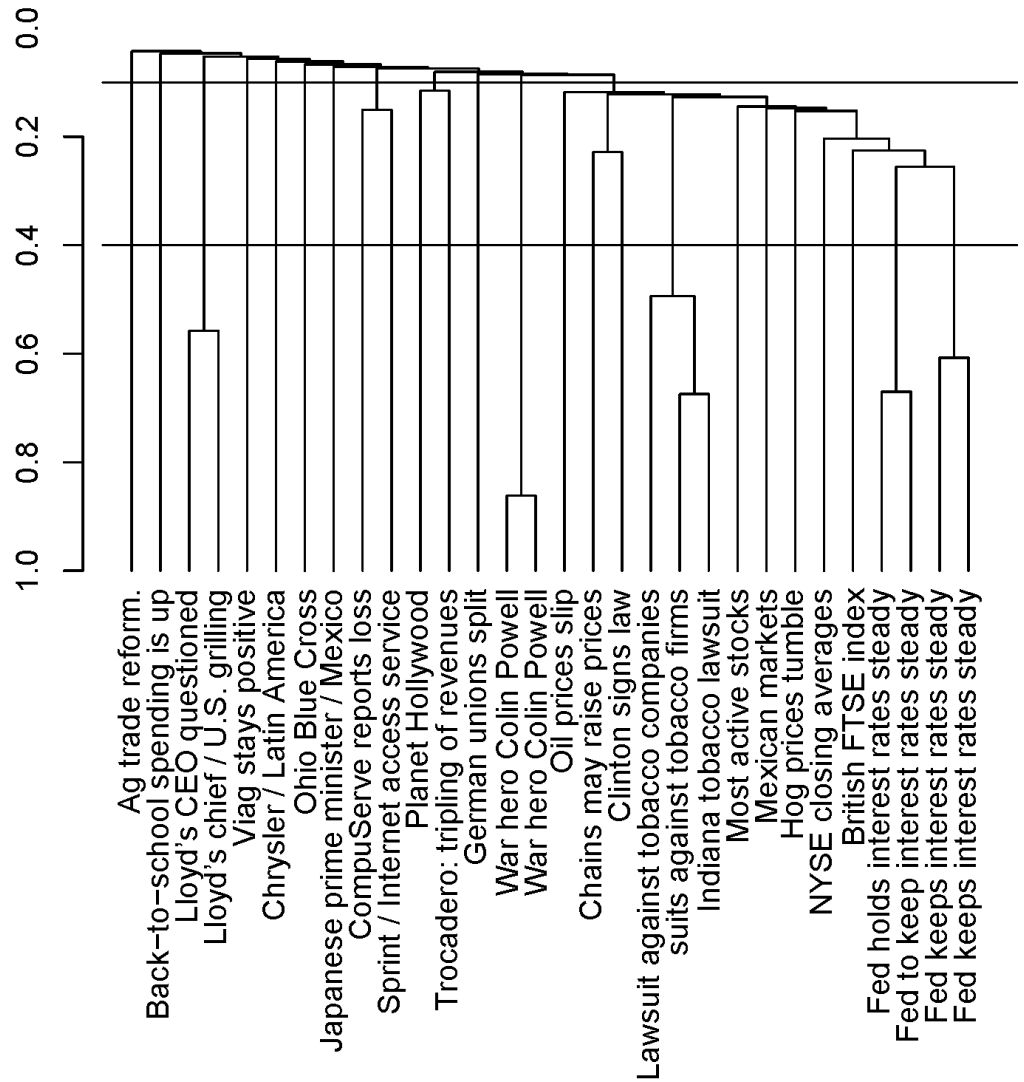
- Build a tree-based hierarchical taxonomy (*dendrogram*) from a set of documents

- One approach: recursive application of a partitional clustering algorithm

Dendogram: Hierarchical Clustering

- Clustering of the documents is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.





► **Figure 17.1** A dendrogram of a single-link clustering of 30 documents from Reuters-RCV1. Two possible cuts of the dendrogram are shown: at 0.4 into 24 clusters and at 0.1 into 12 clusters.

Hierarchical Clustering algorithms

- **Agglomerative (bottom-up):**
  - Start with each document being a single cluster.
  - Eventually all documents belong to the same cluster.
- **Divisive (top-down):**
  - Start with all documents belong to the same cluster.
  - Eventually each node forms a cluster on its own.
- Does not require the number of clusters ***k*** in advance
- Needs a termination/readout condition

Hierarchical Agglomerative Clustering (HAC)

- Starts with all documents in a separate cluster
  - then repeatedly joins the two clusters that are most similar until there is only one cluster.
- The history of merging forms a binary tree or hierarchy.

HAC Algorithm

Start with all documents in their own cluster.
Until there is only one cluster:
    Among the current clusters, determine the two clusters, c_i and c_j, that are most similar.
    Replace c_i and c_j with a single cluster $c_i \cup c_j$

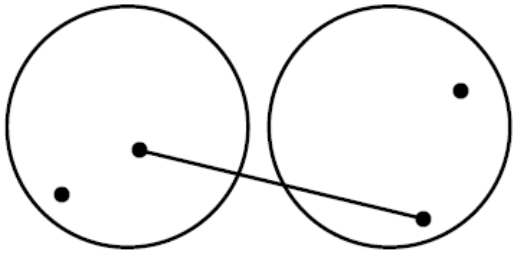
SIMPLEHAC$(d_1, \ldots, d_N)$

1 **for** $n \leftarrow 1$ **to** N
2 **do for** $i \leftarrow 1$ **to** N
3 **do** $C[n][i] \leftarrow \text{SIM}(d_n, d_i)$
4 $I[n] \leftarrow 1$ *(keeps track of active clusters)*
5 $A \leftarrow []$ *(assembles clustering as a sequence of merges)*
6 **for** $k \leftarrow 1$ **to** $N - 1$
7 **do** $\langle i, m \rangle \leftarrow \arg\max_{\{\langle i,m \rangle : i \neq m \wedge I[i]=1 \wedge I[m]=1\}} C[i][m]$
8 A.APPEND$(\langle i, m \rangle)$ *(store merge)*
9 **for** $j \leftarrow 1$ **to** N
10 **do** $C[i][j] \leftarrow \text{SIM}(i, m, j)$
11 $C[j][i] \leftarrow \text{SIM}(i, m, j)$
12 $I[m] \leftarrow 0$ *(deactivate cluster)*
13 **return** A

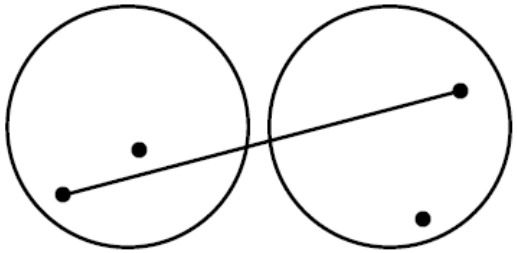
Closest pair of clusters

- Many variants to defining closest pair of clusters
- Single-link clustering
  - Similarity of their most similar members (single-link)
- Complete-link clustering
  - Similarity of their most dissimilar members (“furthest” points)
- Centroid clustering
  - Similarity of their centroids
  - i.e., average similarity of all pairs of documents from different clusters
- Group average-link
  - Average of similarities of all pairs

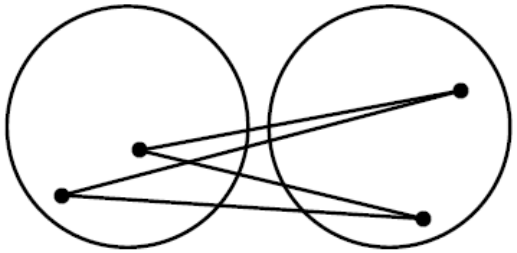
Closest pair of clusters



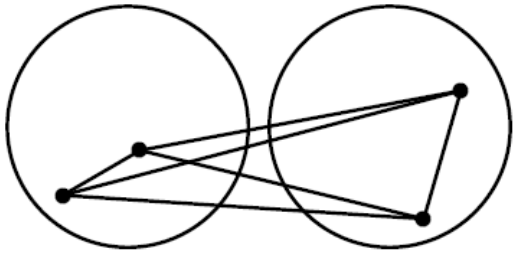
(a) single-link: maximum similarity



(b) complete-link: minimum similarity



(c) centroid: average inter-similarity



(d) group-average: average of all similarities

Single Link Agglomerative Clustering

- Use maximum similarity of pairs:

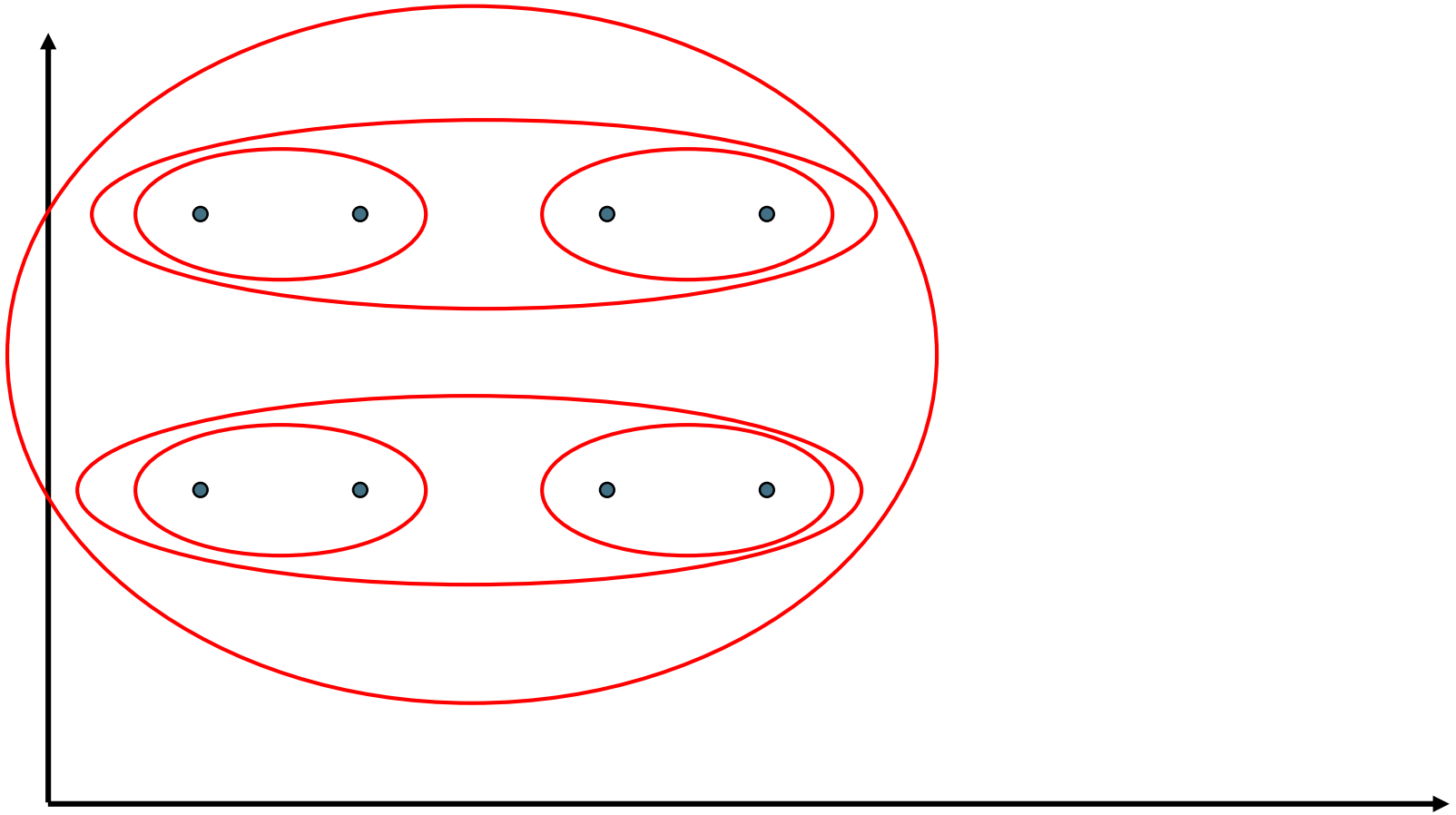
$$sim(c_i, c_j) = \max_{x \in c_i, y \in c_j} sim(x, y)$$

- Can result in "straggly" (long and thin) clusters due to chaining effect.

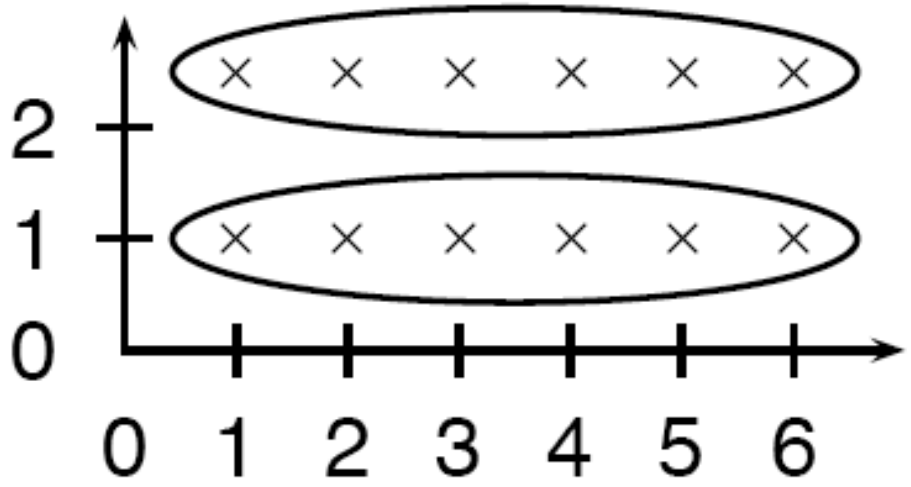
- After merging c_i and c_j, the similarity of the resulting cluster to another cluster, c_k, is:

$$sim((c_i \cup c_j), c_k) = \max(sim(c_i, c_k), sim(c_j, c_k))$$

Single Link Example



Single-link: Chaining



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

Complete Link Agglomerative Clustering

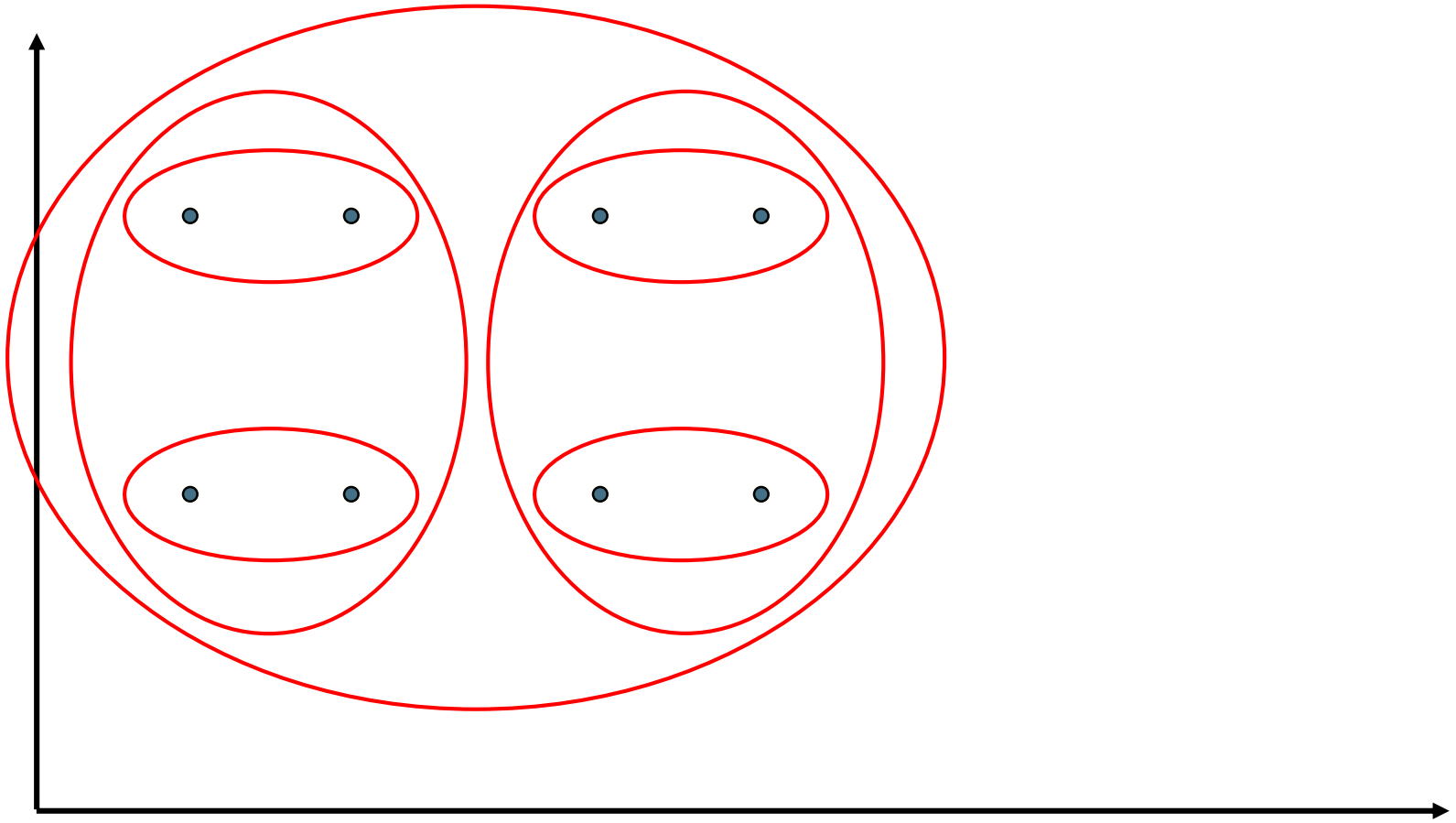
- Use minimum similarity of pairs:

$$sim(c_i, c_j) = \min_{x \in c_i, y \in c_j} sim(x, y)$$

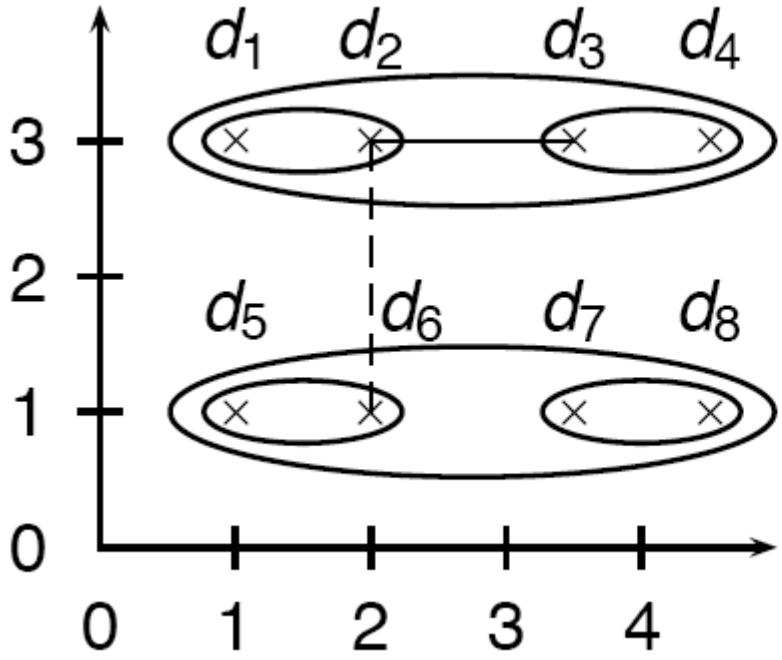
- Makes "tighter," spherical clusters that are typically preferable.
- After merging c_i and c_j, the similarity of the resulting cluster to another cluster, c_k, is:

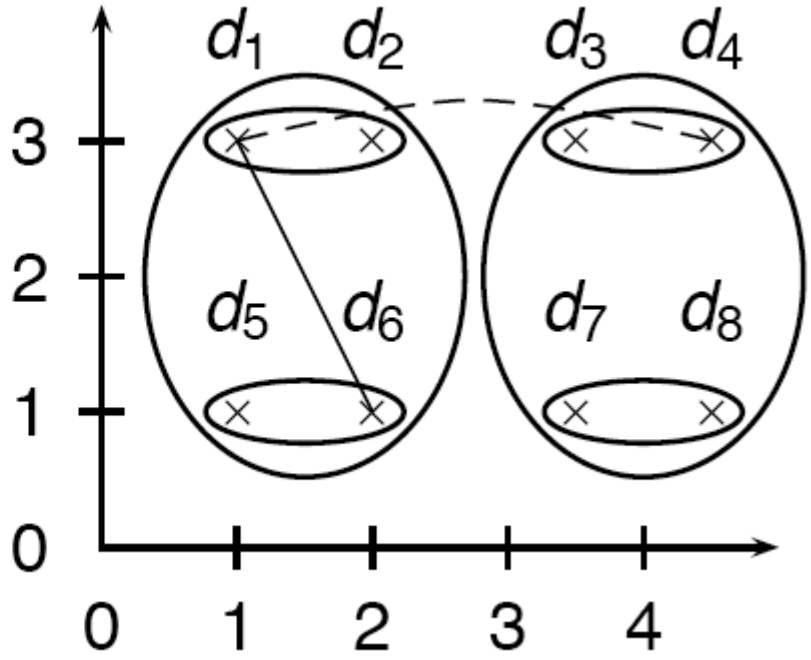
$$sim((c_i \cup c_j), c_k) = \min(sim(c_i, c_k), sim(c_j, c_k))$$

Complete Link Example



Single vs. Complete Link





Computational Complexity

- In the first iteration, all HAC methods need to compute similarity of all pairs of n individual instances (documents) which is $O(n^2)$.
- In each of the subsequent $n-1$ merging iterations, it must compute the distance between the most recently created cluster and all other existing clusters.
- In order to maintain an overall $O(n^2)$ performance, computing similarity to each other cluster must be done in constant time.
  - $O(n^3)$ if done naively or $O(n^2 \log n)$ if done more cleverly

Flat or hierarchical clustering

- For high efficiency, use flat clustering
- For deterministic (same) 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)

Labeling

Major issue - labeling

- After clustering algorithm finds clusters - how can they be useful to the end user?
- Need simple label for each cluster
  - In search results, say “Animal” or “Car” in the ***jaguar*** example.
  - In topic trees (Yahoo), need navigational cues.
    - Often done by hand, a posteriori.

Ideas?

- Use metadata like Titles
- Use the medoid (document) itself – Title
- Top-terms (most frequent)
  - Stop-words, duplicates
- Most distinguishing terms (in my cluster, not in your cluster)

How to Label Clusters

- Show titles of typical documents
  - Titles are easy to scan
  - Authors create them for quick scanning!
  - But you can only show a few titles which may not fully represent cluster
- Show words/phrases prominent in cluster
  - More likely to fully represent cluster
  - Use distinguishing words/phrases
    - Differential labeling
  - But harder to scan

Labeling

- Common heuristics - list 5-10 most frequent terms in the centroid vector.
  - Drop stop-words; stem.
- Differential labeling by frequent terms
  - Within a collection “Computers”, clusters all have the word ***computer*** as frequent term.
  - Discriminant analysis of centroids.
- Perhaps better: distinctive noun phrase

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
- Any feature selection method can also be used for labeling

Final word

- In clustering, clusters are inferred from the data without human input (unsupervised learning)

- However, in practice, it's a bit less clear: there are many ways of influencing the outcome of clustering: number of clusters, similarity measure, representation of documents, . . .

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 human-defined classification
  - Require the ground truth

External criteria for clustering quality

- Based on a gold standard data set, e.g., the Reuters collection
- 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

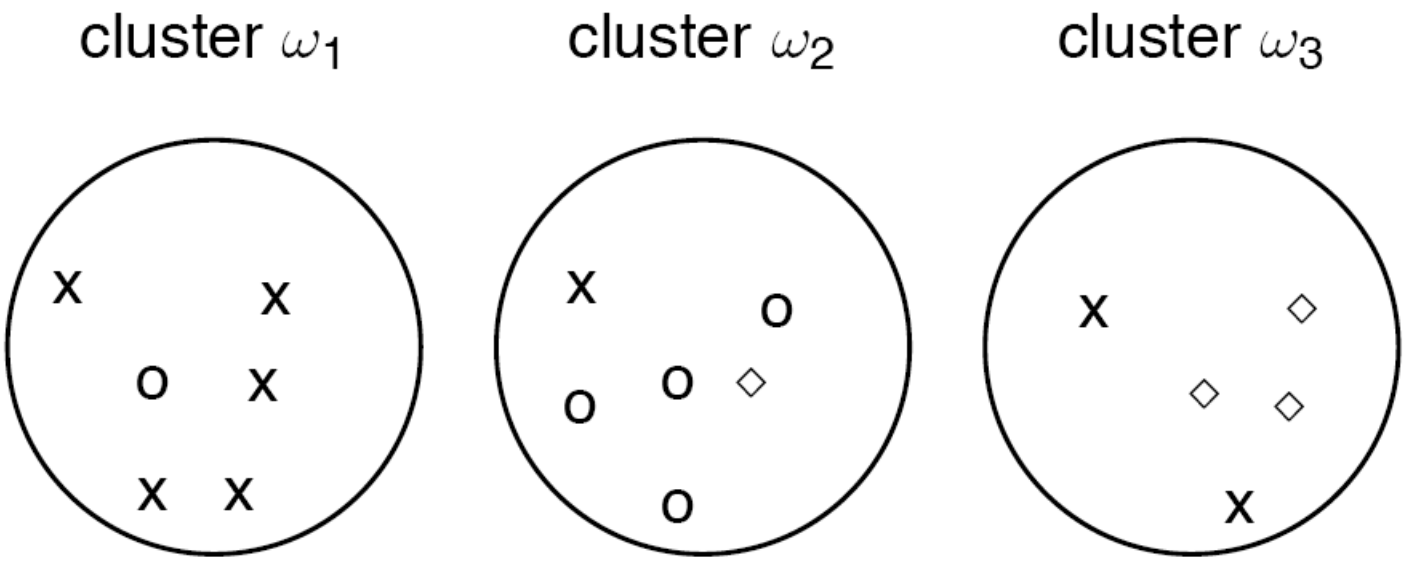
$$\text{purity}(\Omega, \Gamma) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

$\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ is the set of clusters and
$\Gamma = \{c_1, c_2, \ldots, c_J\}$ is the set of classes.

For each cluster ω_k : find class c_j with most members n_{kj} in cluster

Sum all n_{kj} and divide by total number of points

Example

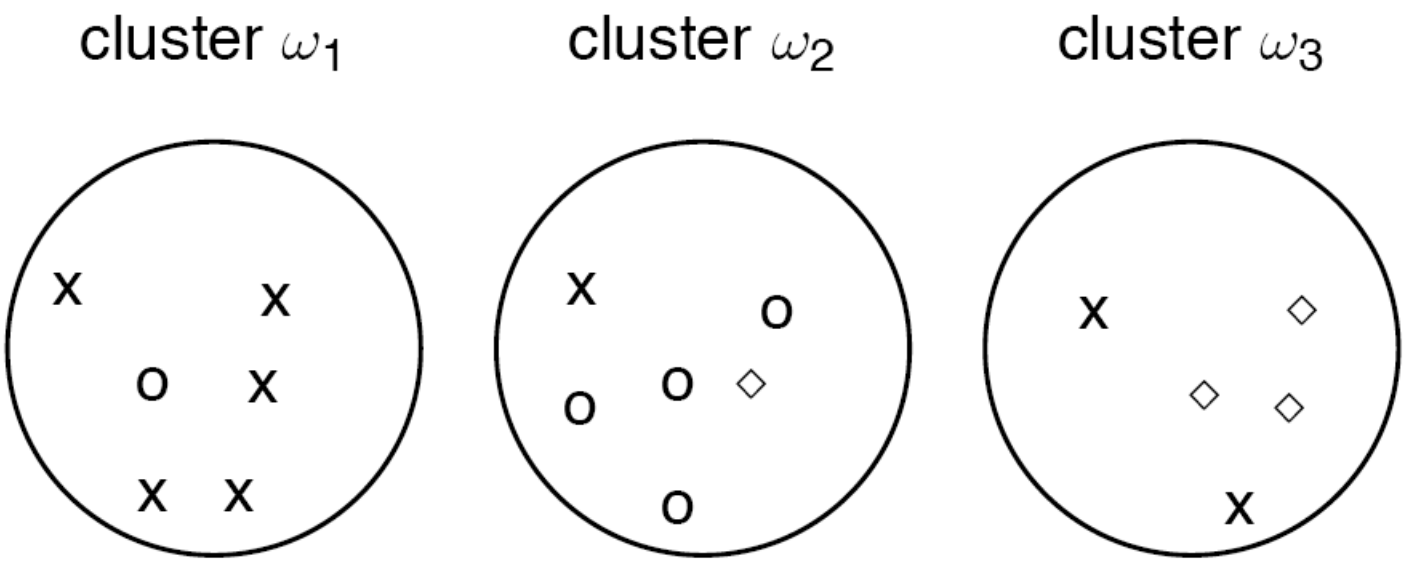


$$\text{good\_docs}(\omega_1) = \max(5,1,0) = 5$$
$$\text{good\_docs}(\omega_2) = \max(1,4,1) = 4$$
$$\text{good\_docs}(\omega_3) = \max(2,0,3) = 3$$
$$\text{purity}(\Omega) = 1/17 \cdot (5+4+3) = 12/17$$

Three other external evaluation measures

- Rand Index
- 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

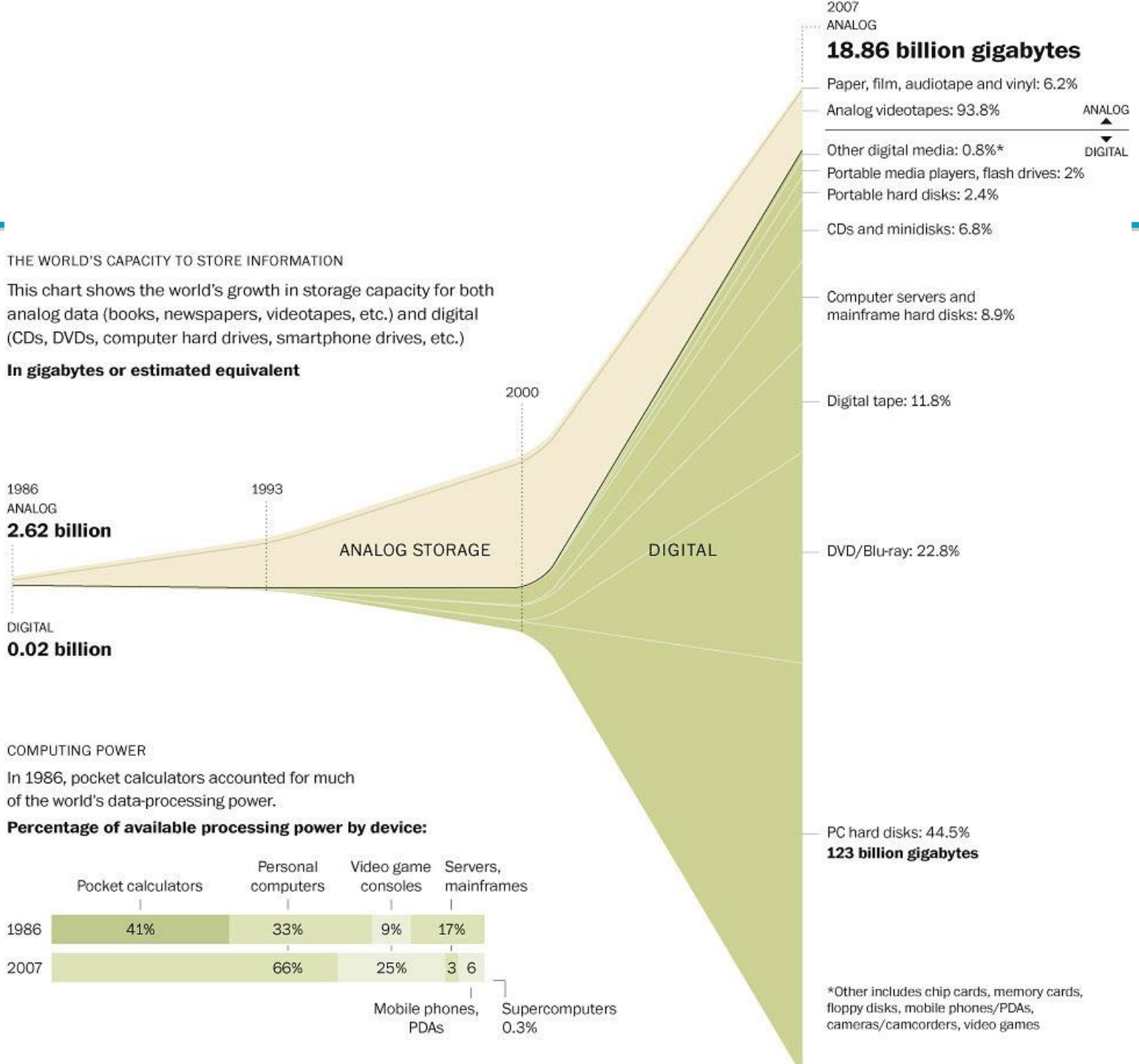
Evaluation results



purity	NMI	RI	F_5
0.71	0.34	0.68	0.46

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

Recommenders


THE WORLD'S CAPACITY TO STORE INFORMATION
This chart shows the world's growth in storage capacity for both analog data (books, newspapers, videotapes, etc.) and digital (CDs, DVDs, computer hard drives, smartphone drives, etc.)
In gigabytes or estimated equivalent
1986
ANALOG
2.62 billion
DIGITAL
0.02 billion
1993
2000
ANALOG STORAGE
DIGITAL
2007
ANALOG
18.86 billion gigabytes
Paper, film, audiotape and vinyl: 6.2%
Analog videotapes: 93.8%
ANALOG
DIGITAL
Other digital media: 0.8%*
Portable media players, flash drives: 2%
Portable hard disks: 2.4%
CDs and minidisks: 6.8%
Computer servers and mainframe hard disks: 8.9%
Digital tape: 11.8%
DVD/Blu-ray: 22.8%
PC hard disks: 44.5%
123 billion gigabytes
COMPUTING POWER
In 1986, pocket calculators accounted for much of the world's data-processing power.
Percentage of available processing power by device:
Pocket calculators
Personal computers
Video game consoles
Servers, mainframes
1986
41%
33%
9%
17%
2007
66%
25%
3
6
Mobile phones, PDAs
Supercomputers 0.3%
*Other includes chip cards, memory cards, floppy disks, mobile phones/PDAs, cameras/camcorders, video games

In 2012


THE MOBILE WEB RECEIVES 217 NEW USERS.
YOUTUBE USERS UPLOAD 48 HOURS OF NEW VIDEO.
EMAIL USERS SEND 204,166,667 MESSAGES.
GOOGLE RECEIVES OVER 2,000,000 SEARCH QUERIES.
FACEBOOK USERS SHARE 684,478 PIECES OF CONTENT.
WORDPRESS USERS PUBLISH 347 NEW BLOG POSTS.
571 NEW WEBSITES ARE CREATED.
CONSUMERS SPEND $272,070 ON WEB SHOPPING.
EVERY MINUTE of the DAY
FOURSQUARE USERS PERFORM 2,083 CHECK-INS.
TWITTER USERS SEND OVER 100,000 TWEETS.
FLICKR

How Many Products Does Amazon Sell?

by Paul Grey on 15 December 2013 in E-Commerce

Amazon.com is the self-styled "Greatest Store on Earth." It's been said that Amazon aims one day to sell everything to everyone.

Exactly how much choice do you have when shopping with Amazon?

Today Amazon sells over 200 million products in the USA, which are categorised into 35 departments. There are almost 5 million items in the Clothing department, almost 20 million in Sports & Outdoors, and over 4 million Office Products. There 7 million items in the Amazon Jewelry department, 24 million in Electronics, 1.4 million products in the Beauty department, 570 thousand Baby products, and 600 thousand Grocery items.

That's in the USA. This table lists my estimates of the number of products offered on the main Amazon websites around the world.

Amazon.com	USA	232 million
Amazon.co.uk	UK	132 million
Amazon.de	Germany	118 million

Apple: iTunes Now Has 20M Songs; Over 16B Downloads

Posted Oct 4, 2011 by ***Leena Rao*** *(@leenarao)*

At today's Apple iPhone event, newly appointed CEO Tim Cook revealed some staggering numbers on Apple's music downloads and songs in iTunes. iTunes now offers 20 million songs, which is up from 200,000 when iTunes was first launched eight years ago.

iTunes is the number one music store in the world with over 16 billion songs downloaded. It looks like Apple has seen about a billion downloads in the past four months, as the company revealed 15 billion song downloads in June. At the time, Apple revealed that it 225 million credit card accounts listed with iTunes.

Interestingly, streaming service Spotify has around 15 million songs available, which isn't that much less.

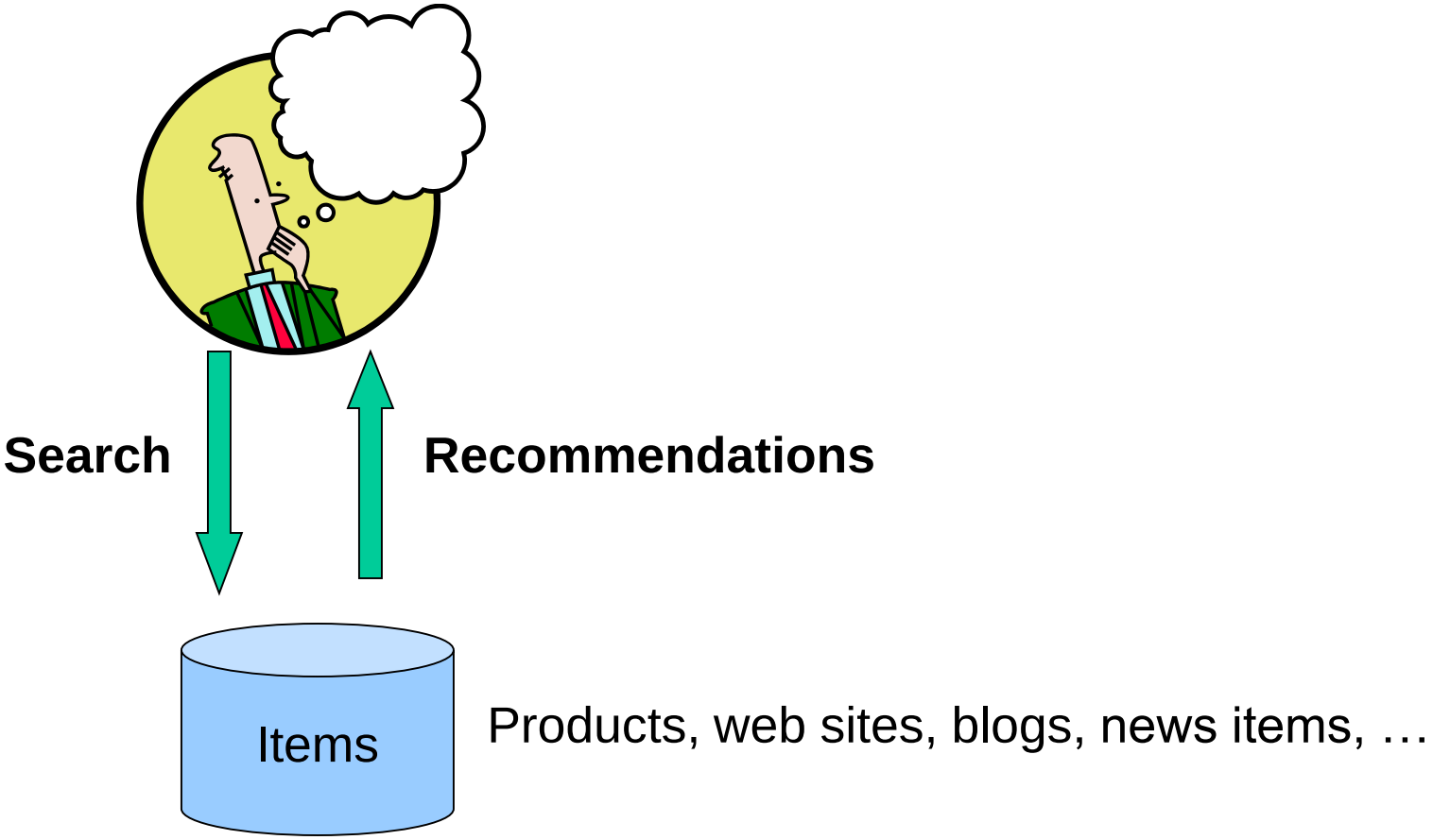
Statistics

Viewership

- More than 1 billion unique users visit YouTube each month
- Over 6 billion hours of video are watched each month on YouTube—that's almost an hour for every person on Earth
- 100 hours of video are uploaded to YouTube every minute
- 80% of YouTube traffic comes from outside the US
- YouTube is localized in 61 countries and across 61 languages
- According to Nielsen, YouTube reaches more US adults ages 18-34 than any cable network
- Millions of subscriptions happen each day. The number of people subscribing daily is up more than 3x since last year, and the number of daily subscriptions is up more than 4x since last year

In 2014

Recommendations



Amazon.com: Online Shopping for Electronics, Apparel, Computers, books, DVDs & more

www.amazon.com

Reader

amazon Prime

James's Amazon.com | Today's Deals | Gift Cards | Sell | Help

Back-to-School Deals
Sponsored by Lysol
› Shop now

Shop by Department | Search | All | Go | Hello, James Your Account | Your Prime | 0 Cart | Wish List

Instant Video | Digital Music Store | Cloud Drive | Kindle | Appstore for Android | Digital Games & Software | Audible Audiobooks

eTextbooks for your iPad

Save up to 80% with eTextbooks and carry less

› Learn more

Top Picks: Women's Denim | Really Great Toys | Off to College

Building on a Family Legacy

Kathryn Parish's son added Internet sales to build on his family's six-decade tradition of delighting kids and parents with quality toys.

› "Mom got a big kick out of it"

› Shop Really Great Toys

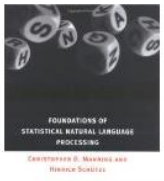
One of thousands of small businesses thriving because of Amazon customers.

Related to Items You've Viewed

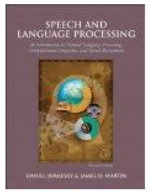
You viewed | Customers who viewed this also viewed

Introduction to Information Retrieval
› Hinrich Schütze, Christopher D. Manning, Prabhakar Raghavan
Hardcover
★★★★½ (21)
~~$69.00~~ $57.42

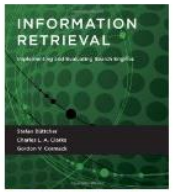
Taming Text: How to Find, Organize...
Grant S. Ingersoll, Thomas S. Morton, ...
Paperback
★★★★★ (9)
~~$44.99~~ $31.65



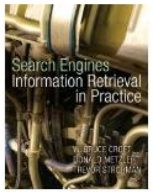
Foundations of Statistical Natural...
› Hinrich Schütze, Christopher D. Manning
Hardcover
★★★★½ (18)
~~$95.00~~ $85.00



Speech and Language Processing, 2nd...
Daniel Jurafsky, James H. Martin
Hardcover
★★★★☆ (15)
~~$177.20~~ $141.37



Information Retrieval: Implementing...
Stefan Buettcher, Charles L. A...
Hardcover
★★★★★ (4)
~~$62.00~~ $40.00

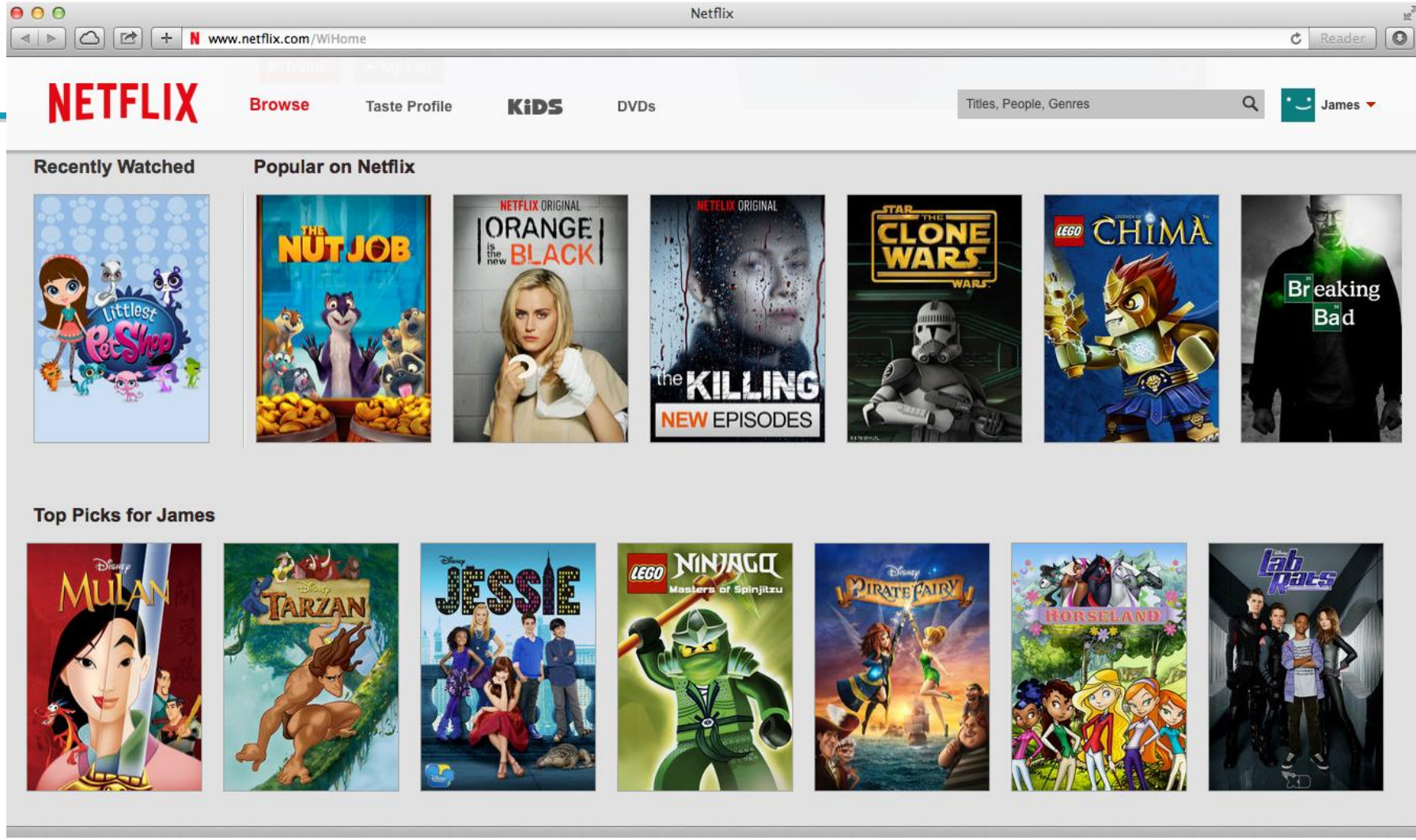


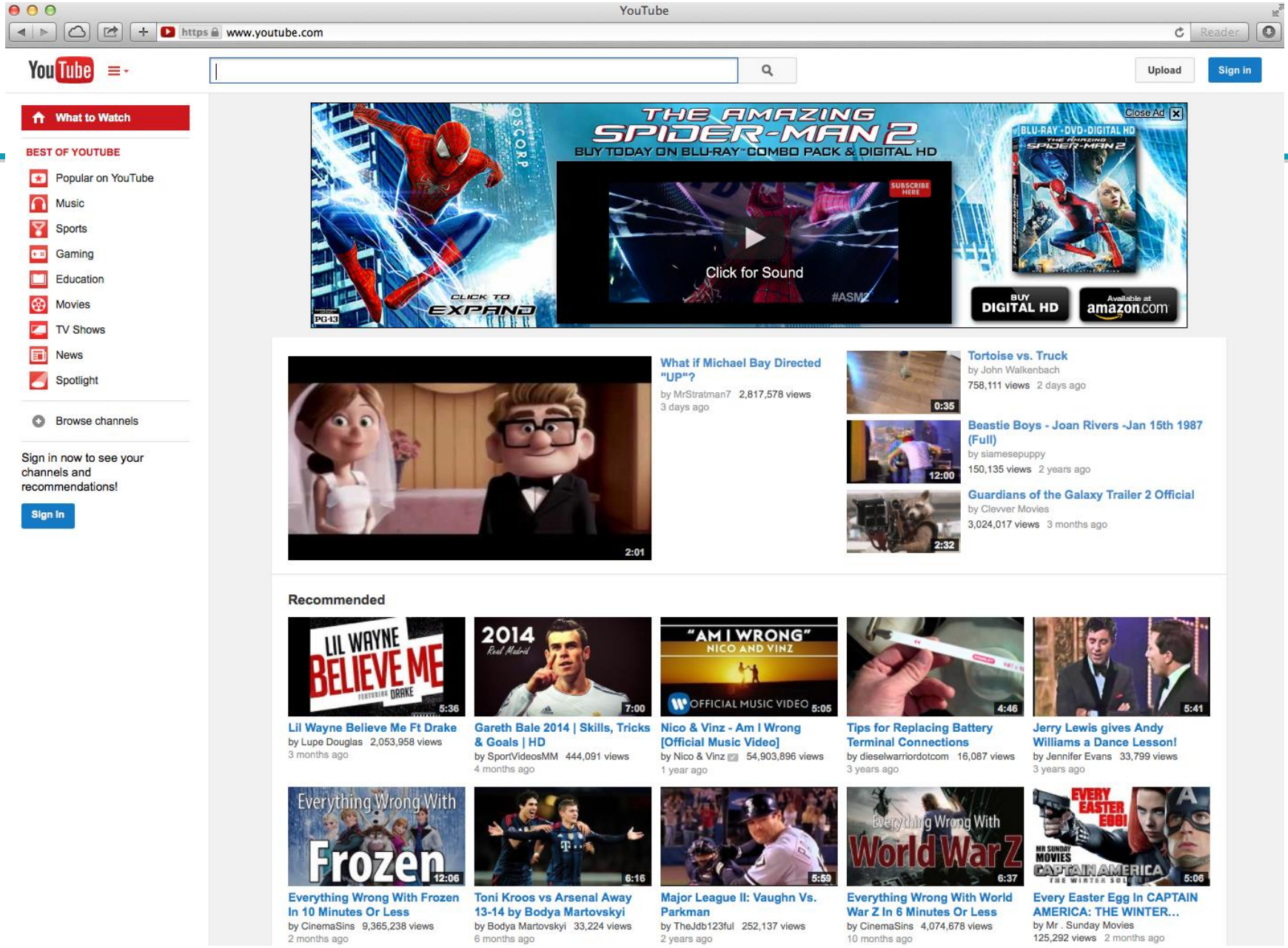
Search Engines: Information Retrieval...
Bruce Croft, Donald Metzler, Trevor...
Hardcover
★★★★☆ (9)
~~$133.80~~ $111.84

› View or edit your browsing history

Best Sellers

Sports & Outdoors : Golf Clubs


Netflix
www.netflix.com/WiHome
Reader
NETFLIX
Browse
Taste Profile
KiDS
DVDs
Titles, People, Genres
James
Recently Watched
Popular on Netflix
Littlest Pet Shop
THE NUT JOB
NETFLIX ORIGINAL
ORANGE is the new BLACK
NETFLIX ORIGINAL
the KILLING
NEW EPISODES
STAR WARS THE CLONE WARS
LEGO CHIMA
Breaking Bad
Top Picks for James
Disney MULAN
Disney TARZAN
Disney JESSIE
LEGO NINJAGO
Masters of Spinjitzu
Disney PIRATE FAIRY
HORSELAND
Lab Rats


YouTube
https www.youtube.com
Reader
YouTube
Upload
Sign in
What to Watch
BEST OF YOUTUBE
Popular on YouTube
Music
Sports
Gaming
Education
Movies
TV Shows
News
Spotlight
Browse channels
Sign in now to see your channels and recommendations!
Sign In
THE AMAZING SPIDER-MAN 2
BUY TODAY ON BLU-RAY™ COMBO PACK & DIGITAL HD
Close Ad
CLICK TO EXPAND
Click for Sound
#ASM2
BLU-RAY · DVD · DIGITAL HD
BUY DIGITAL HD
Available at amazon.com
2:01
What if Michael Bay Directed "UP"?
by MrStratman7 2,817,578 views
3 days ago
0:35
Tortoise vs. Truck
by John Walkenbach
758,111 views 2 days ago
12:00
Beastie Boys - Joan Rivers -Jan 15th 1987 (Full)
by siamesepuppy
150,135 views 2 years ago
2:32
Guardians of the Galaxy Trailer 2 Official
by Clevver Movies
3,024,017 views 3 months ago
Recommended
LIL WAYNE BELIEVE ME FEATURING DRAKE
5:36
Lil Wayne Believe Me Ft Drake
by Lupe Douglas 2,053,958 views
3 months ago
2014 Real Madrid
7:00
Gareth Bale 2014 | Skills, Tricks & Goals | HD
by SportVideosMM 444,091 views
4 months ago
"AM I WRONG" NICO AND VINZ
OFFICIAL MUSIC VIDEO 5:05
Nico & Vinz - Am I Wrong [Official Music Video]
by Nico & Vinz 54,903,896 views
1 year ago
4:46
Tips for Replacing Battery Terminal Connections
by dieselwarriordotcom 16,087 views
3 years ago
5:41
Jerry Lewis gives Andy Williams a Dance Lesson!
by Jennifer Evans 33,799 views
3 years ago
Everything Wrong With Frozen
12:06
Everything Wrong With Frozen In 10 Minutes Or Less
by CinemaSins 9,365,238 views
2 months ago
6:16
Toni Kroos vs Arsenal Away 13-14 by Bodya Martovskyi
by Bodya Martovskyi 33,224 views
6 months ago
5:59
Major League II: Vaughn Vs. Parkman
by TheJdb123ful 252,137 views
2 years ago
Everything Wrong With World War Z
6:37
Everything Wrong With World War Z In 6 Minutes Or Less
by CinemaSins 4,074,678 views
10 months ago
EVERY EASTER EGG!
MR SUNDAY MOVIES
CAPTAIN AMERICA
5:06
Every Easter Egg In CAPTAIN AMERICA: THE WINTER...
by Mr . Sunday Movies
125,292 views 2 months ago

Magic!

Similar Artists

▶ Play similar artists

Rixton
68,883 listeners

Rixton are a British Pop/Rock/R&B band from Manchester, UK. Members of Rixton are lead singer and rhythm guitarist Jake Roche, bass/keys, backing vocalist Danny Wilkin, lead guitarist and backing vocalist, Charley Bagnall, and drummer Lewi Morgan.

Meghan Trainor
28,437 listeners

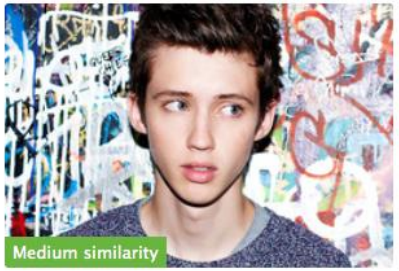
Meghan was in the band Island Fusion for 4 years. She sang lead and back-up vocals, performed keyboards, guitar, and auxiliary percussion. The band performed many events and fund-raisers on Nantucket Island, including opening for Jamaican artist Beenie Man.

Katy Tiz
10,534 listeners

I sing,in the shower. Maybe one day I'll get a real job. Monsters made me do it. @KatyTizMusic "I'm always the loud one," smiles Katy Tiz.

Echosmith
48,925 listeners

Jacquie Lee
22,458 listeners



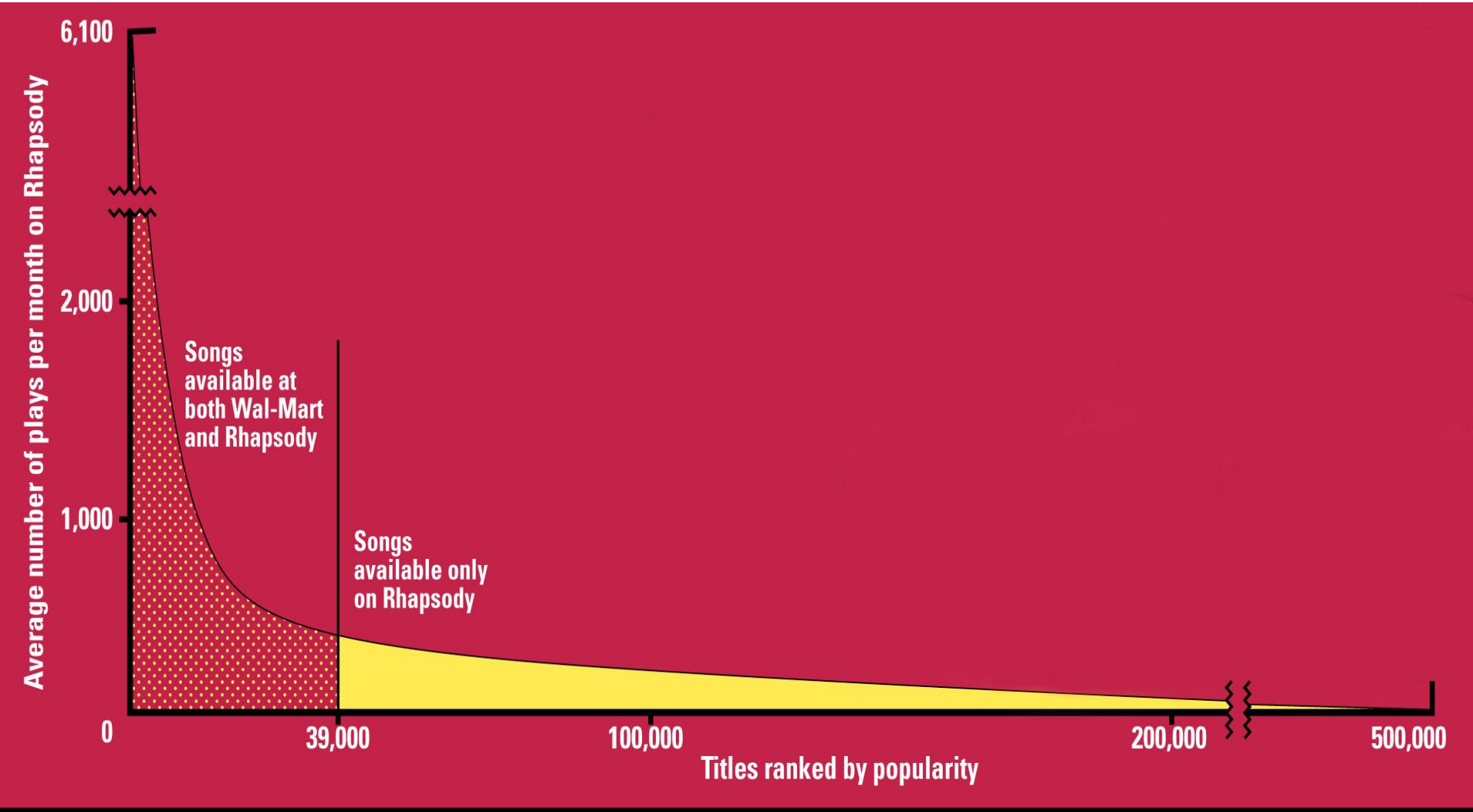
Troye Sivan
14,333 listeners

Other examples of recommenders?

From Scarcity to Abundance

- Shelf space is a scarce commodity for traditional retailers
  - Also: TV networks, movie theaters,…
- Web enables near-zero-cost dissemination of information about products
  - From scarcity to abundance
- More choice necessitates better filters
  - Recommendation engines
  - How Into Thin Air made Touching the Void a bestseller
    - http://www.wired.com/wired/archive/12.10/tail.html

The Long Tail

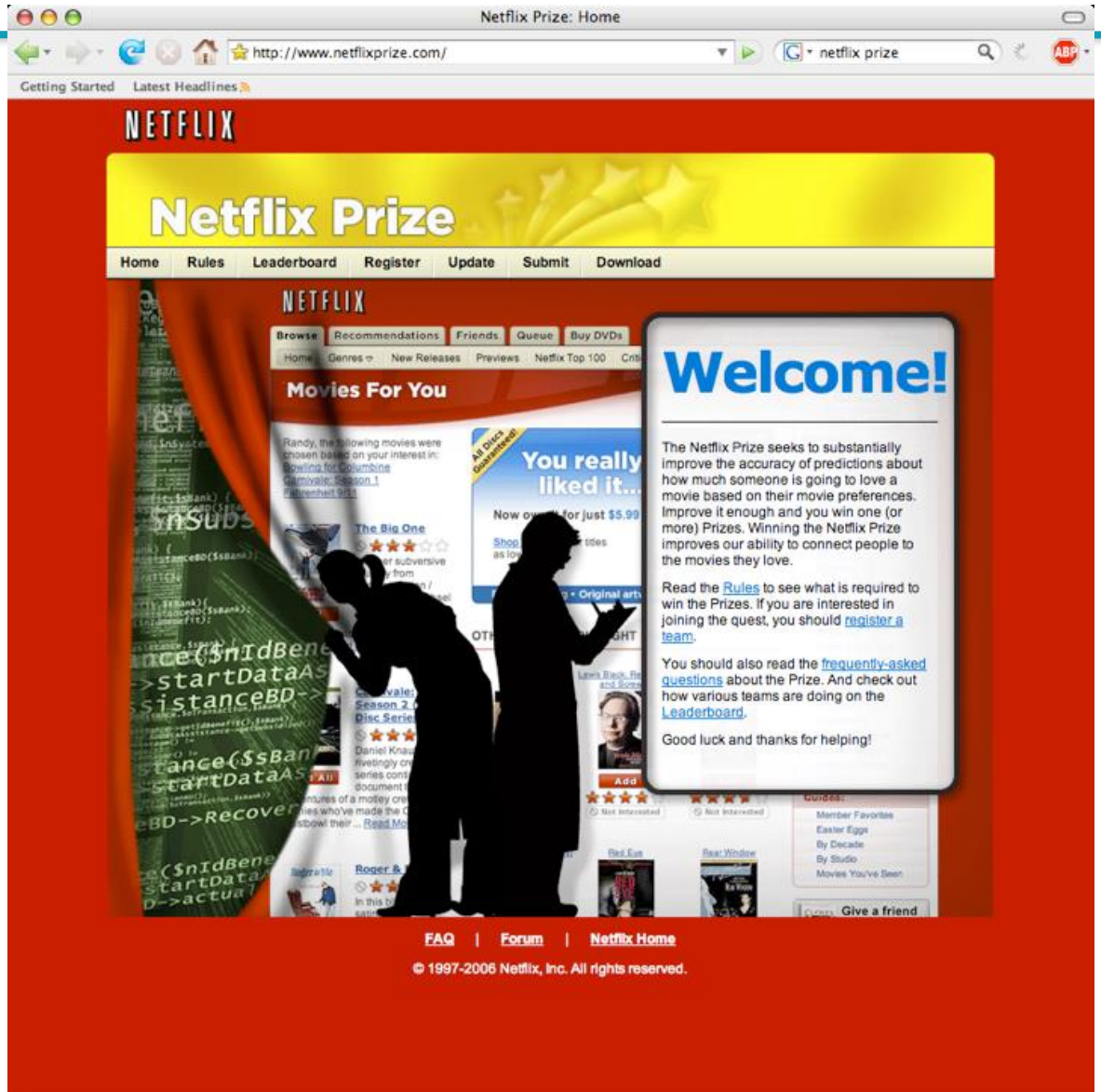


Source: Chris Anderson (2004)

Recommendation Types

- Editorial and hand curated
  - List of favorites
  - Lists of “essential” items
- Simple aggregates
  - Top 10, Most Popular, Recent Uploads
- Tailored to individual users
  - Amazon, Netflix, …

$$$


Netflix Prize: Home
http://www.netflixprize.com/
netflix prize
Getting Started
Latest Headlines
NETFLIX
Netflix Prize
Home
Rules
Leaderboard
Register
Update
Submit
Download
Browse
Recommendations
Friends
Queue
Buy DVDs
Movies For You
Welcome!
The Netflix Prize seeks to substantially improve the accuracy of predictions about how much someone is going to love a movie based on their movie preferences. Improve it enough and you win one (or more) Prizes. Winning the Netflix Prize improves our ability to connect people to the movies they love.
Read the Rules to see what is required to win the Prizes. If you are interested in joining the quest, you should register a team.
You should also read the frequently-asked questions about the Prize. And check out how various teams are doing on the Leaderboard.
Good luck and thanks for helping!
FAQ | Forum | Netflix Home
© 1997-2006 Netflix, Inc. All rights reserved.

Formal Model

- X = set of Customers
- S = set of Items

- Utility function u: $X \times S \rightarrow R$
  - R = set of ratings
  - R is a totally ordered set
  - e.g., 0-5 stars, real number in [0,1]

Utility Matrix

	The Lego Movie	The Fault in Our Stars	Guardians of the Galaxy	Star Wars
Alice	1		0.2	
Bob		0.5		0.3
Carol	0.2		1	
David				0.4

Key Problems

- Gathering “known” ratings for matrix
- Extrapolate unknown ratings from the known ones
  - Mainly interested in high unknown ratings
  - Don’t care about finding what you **don’t like**, but rather what you like
- Evaluating extrapolation methods
  - How do we know if we’ve done a good job?

Gathering Ratings

- Explicit
  - Ask people to rate items
  - Doesn't work well in practice – people can't be bothered
- Implicit
  - Learn ratings from user actions
    - E.g., purchase implies high rating
  - What about low ratings?

(2) Extrapolating Utilities

- Key problem: Utility matrix *U* is sparse
  - Most people have not rated most items
  - Cold start:
    - New items have no ratings
    - New users have no history
- Three main approaches
  - Content-based
  - Collaborative
  - Latent factor based

Content-based Recommender Systems

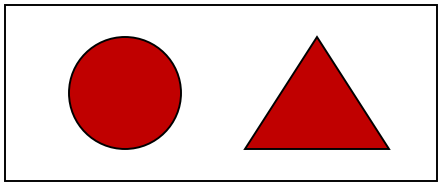
Content-based Recommendations

- **Main idea**: recommend items to customer ***x*** similar to previous items rated highly by ***x***
- Movie recommendations
  - recommend movies with same actor(s), director, genre, …
- Websites, blogs, news
  - recommend other sites with “similar” content

Plan of Action

likes

Item profiles

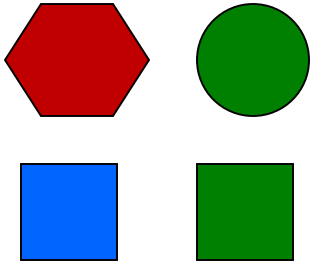


build

Red
Circles
Triangles

User profile

match



recommend

Item Profiles

- For each item, create an item profile
- Profile is a set of features (vectors!)
  - Movies: author, title, actor, director,…
  - Text: Set of “important” words in document
- How to pick important features?
  - Usual heuristic is TF-IDF

Sidenote: TF-IDF

f_{ij} = frequency of term (feature) i in doc (item) j

$$TF_{ij} = \frac{f_{ij}}{\max_k f_{kj}}$$

Note: we normalize TF to discount for "longer" documents

n_i = number of docs that mention term i

N = total number of docs

$$IDF_i = \log \frac{N}{n_i}$$

TF-IDF score: $w_{ij} = TF_{ij} \times IDF_i$

Doc/item profile = set of words with highest TF-IDF scores, together with their scores

User Profiles and Prediction

- User profile possibilities:
  - Weighted average of rated item profiles
  - Variation: weight by difference from average rating for item
  - …
- Prediction heuristic
  - Given user profile **x** and item profile **i**, estimate
    - $u(\mathbf{x},\mathbf{i}) = \cos(\mathbf{x},\mathbf{i}) = \mathbf{x}.\mathbf{i}/(|\mathbf{x}||\mathbf{i}|)$

Advantages of Content-based Approach

- No need for data on other users
  - No cold-start or sparsity problems
- Able to recommend to users with unique tastes
- Able to recommend new & unpopular items
  - No first-rater problem
- Can provide explanations of recommended items by listing content-features that caused an item to be recommended

Limitations of content-based approach

- Finding the appropriate features is hard
  - e.g., images, movies, music
- Recommendations for new users
  - How to build a user profile?
- Overspecialization
  - Never recommends items outside user's content profile
  - People might have multiple interests
  - Unable to exploit quality judgments of other users