

Introduction to Data Science

WFAiS UJ, Informatyka Stosowana
I stopień studiów

This lecture is
based on course by E. Fox and C. Guestrin, Univ of Washington

Retrieving documents of interest

- Currently reading article you like
- **Goal:** Want to find similar article



Retrieving documents of interest

Challenges

- How do we measure similarity?
- How do we search over articles?

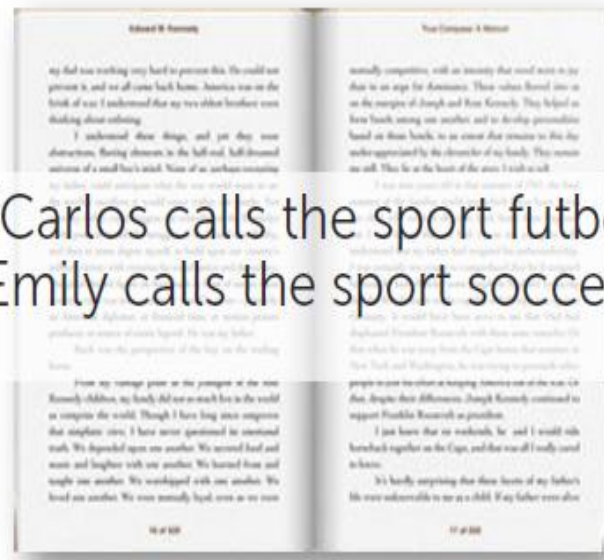


Word count document representation

- Bag of words model
 - Ignore order of words
 - Count # of instances of each word in vocabulary

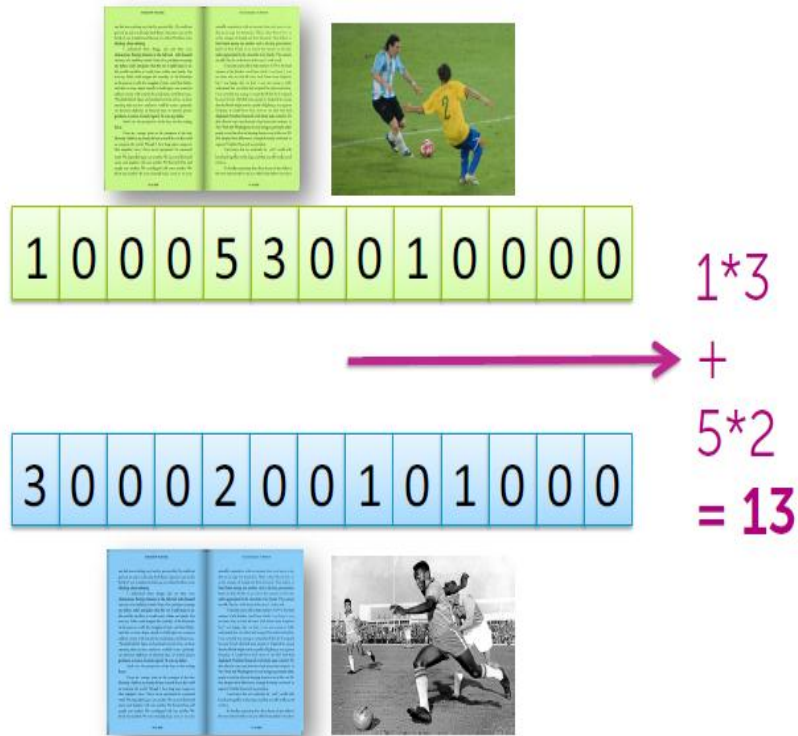


"Carlos calls the sport futbol.
Emily calls the sport soccer."

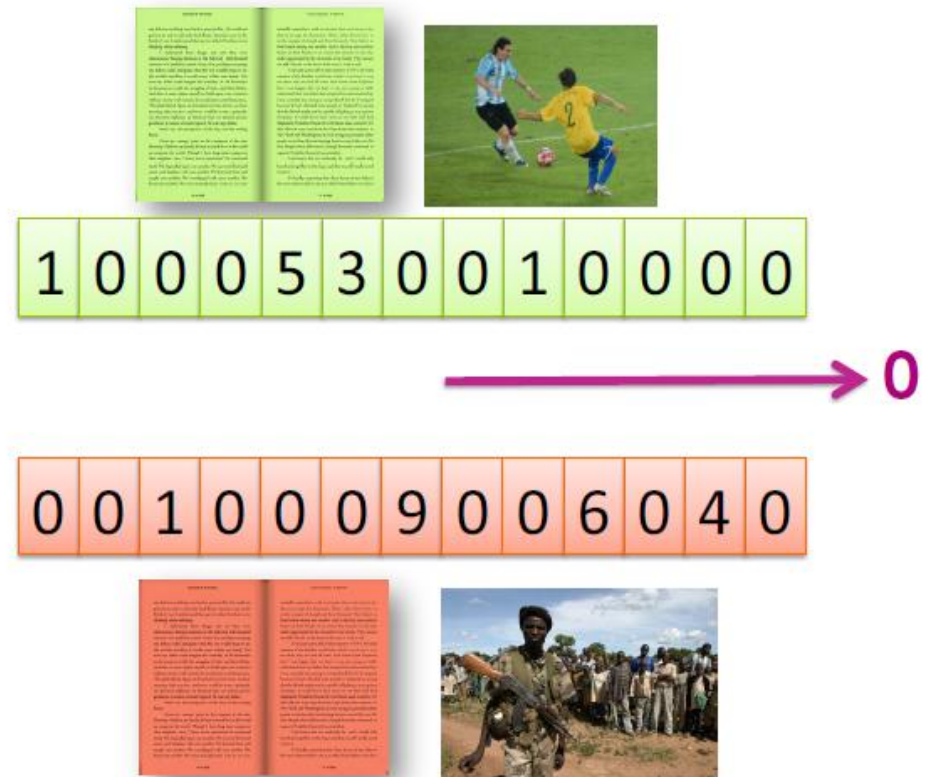


Word count document representation

Measuring similarity

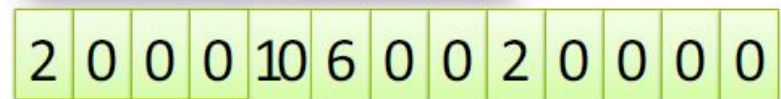
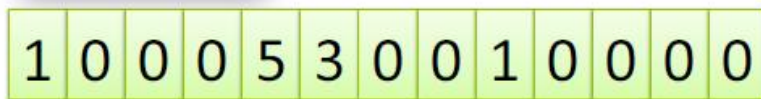


Measuring similarity



Word count document representation

Issues with word counts – Doc length



Similarity = 13



Similarity = 52



Word count document representation

Solution = normalize



1	0	0	0	5	3	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---

$$\sqrt{(1^2 + 5^2 + 3^2 + 1^2)}$$

1				5	3			1				
/	0	0	0	/	/	0	0	/	0	0	0	0
6				6	6			6				

Prioritizing important words

Issues with word counts – Rare words



Common words in doc: "the", "player", "field", "goal"

Dominate rare words like: "futbol", "Messi"

Prioritizing important words

Document frequency

- What characterizes a rare word?
 - Appears infrequently in the corpus
- Emphasize words appearing in few docs
 - Equivalently, discount word w based on
of docs containing w in corpus

Prioritizing important words

Important words

- Do we want only rare words to dominate???
- What characterizes an important word?
 - Appears frequently in document (common locally)
 - Appears rarely in corpus (rare globally)
- Trade off between local frequency and global rarity

TF-IDF document representation

- Term frequency – inverse document frequency (tf-idf)

- Term frequency



- Inverse document frequency



$$\log \frac{\# \text{ docs}}{1 + \# \text{ docs using word}}$$



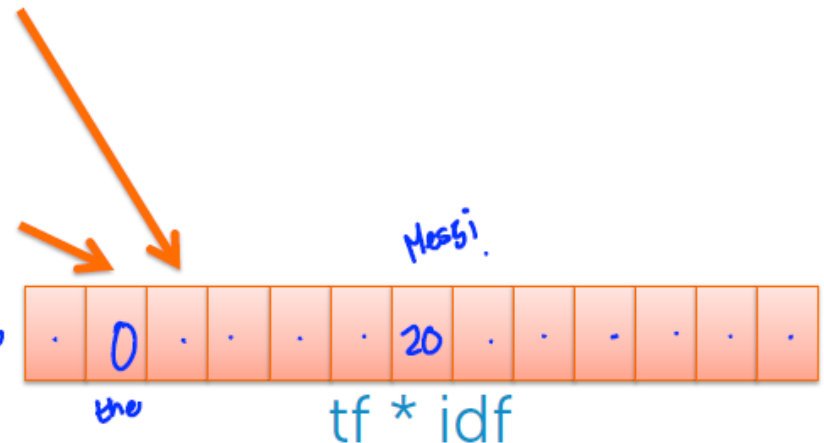
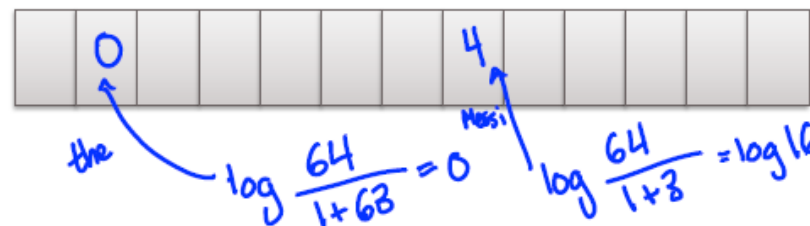
TF-IDF document representation

TF-IDF document representation

- Term frequency – inverse document frequency (tf-idf)
- Term frequency



- Inverse document frequency



Retrieving similar documents

Nearest neighbor search

- Query article:



- Corpus:



- **Specify:** Distance metric
- **Output:** Set of most similar articles



Retrieving similar documents

1 – Nearest neighbor

- **Input:** Query article 
- **Output:** *Most* similar article
- Algorithm:
 - Search over each article  in corpus
 - Compute $s = \text{similarity}(\text{query article}, \text{article})$
 - If $s > \text{Best_s}$, record  =  and set $\text{Best_s} = s$
 - Return 

Retrieving similar documents

k – Nearest neighbor

- **Input:** Query article 
- **Output:** *List of k* similar articles



Structure documents by topics

What if some of the labels are known?

- Training set of labeled docs



SPORTS



WORLD NEWS



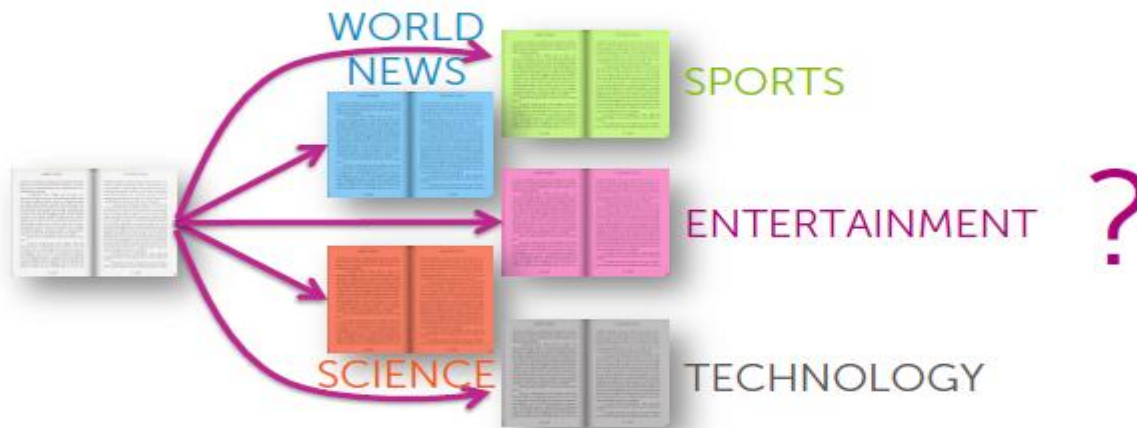
ENTERTAINMENT



SCIENCE

Structure documents by topics

Multiclass classification problem



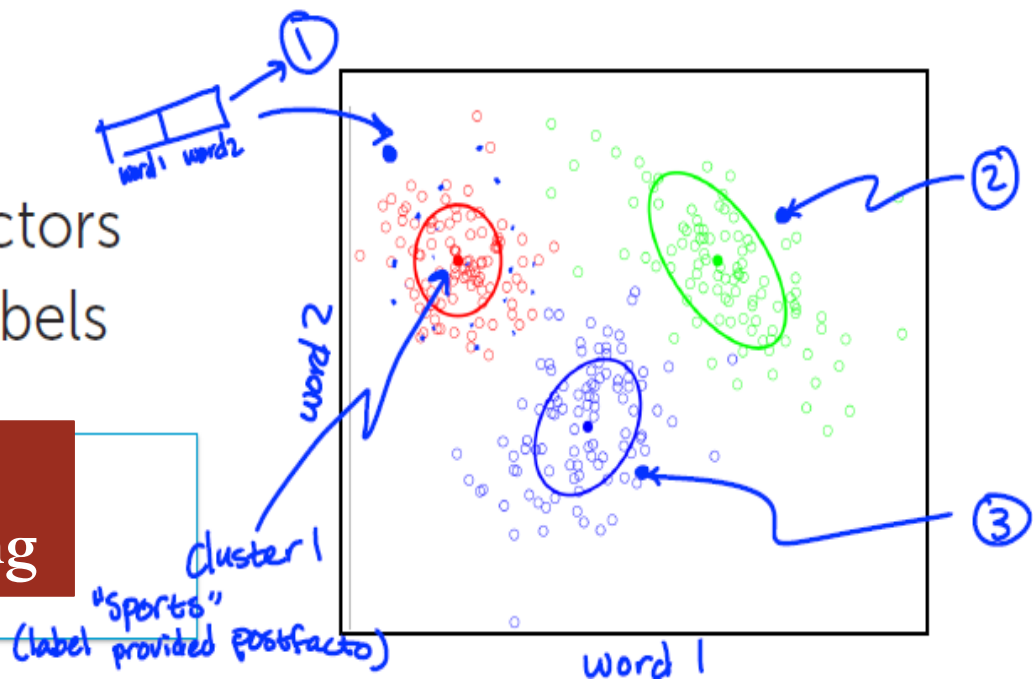
Labels provided: case of supervised learning problem

Clustering

- No labels provided
- Want to uncover cluster structure

- **Input:** docs as vectors
- **Output:** cluster labels

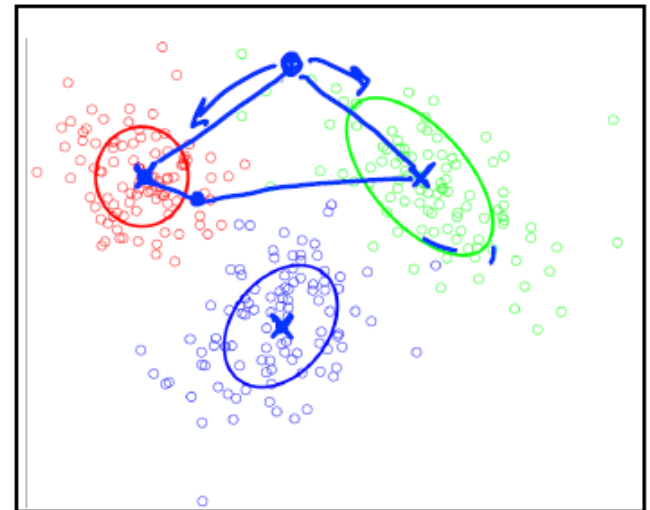
No labels provided
unsupervised learning



Clustering

What defines a cluster?

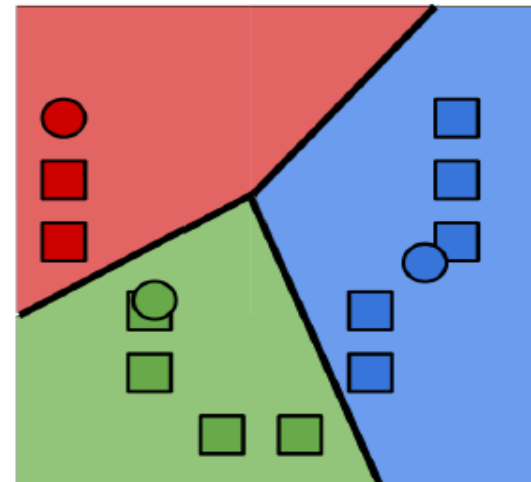
- Cluster defined by center & shape/spread
- Assign observation (doc) to cluster (topic label)
 - Score under cluster is higher than others
 - Often, just more similar to assigned cluster center than other cluster centers



Clustering

k-means algorithm

0. Initialize cluster centers
1. Assign observations to closest cluster center
2. Revise cluster centers as mean of assigned observations
3. Repeat 1.+2. until convergence



Examples

Clustering images

- For search, group as:
 - Ocean
 - Pink flower
 - Dog
 - Sunset
 - Clouds
 - ...



Examples

Products on Amazon

- Discover product categories from purchase histories

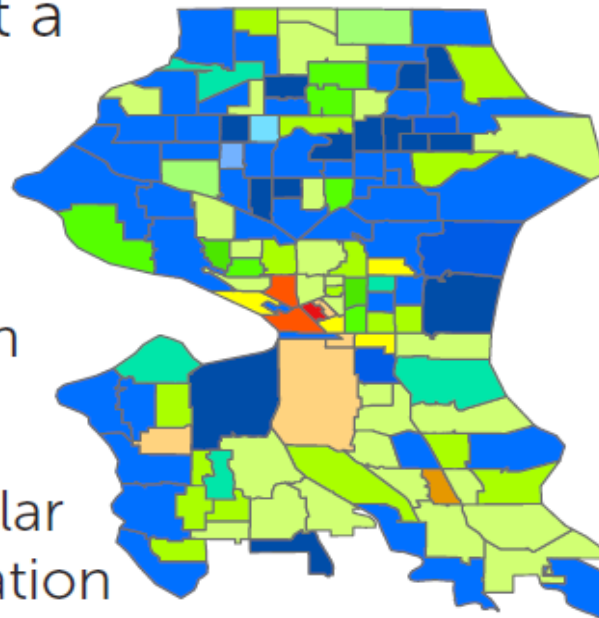


- Or discovering groups of **users**

Examples

Discovering similar neighborhoods

- **Task 1:** Estimate price at a small regional level
- **Challenge:**
 - Only a few (or no!) sales in each region per month
- **Solution:**
 - Cluster regions with similar trends and share information within a cluster

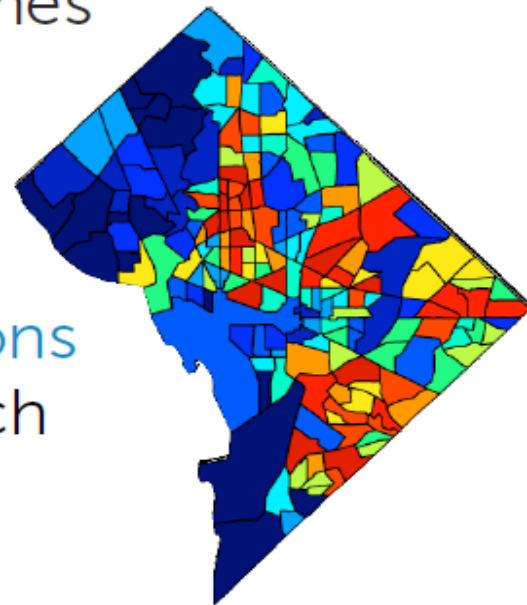


City of Seattle

Examples

Discovering similar neighborhoods

- **Task 2:** Forecast violent crimes to better task police
- Again, **cluster regions** and **share information!**
- Leads to **improved predictions** compared to examining each region independently



Washington, DC

We discussed how to ...

- Describe ways to represent a document (e.g., raw word counts, tf-idf,...)
- Measure the similarity between two documents
- Discuss issues related to using raw word counts
 - Normalize counts to adjust for document length
 - Emphasize important words using tf-idf
- Implement a nearest neighbor search for document retrieval
- Describe the input (unlabeled observations) and output (labels) of a clustering algorithm
- Determine whether a task is supervised or unsupervised
- Cluster documents using k-means (algorithmic details to come...)
- Describe other applications of clustering