Processamento e Recuperação de Informação

Document Clustering and Dimensionality Reduction

Departamento de Engenharia Informática
Instituto Superior Técnico

1º Semestre
2018/2019

Bibliography

- Bing Liu, Web Data Mining - Exploring Hyperlinks, Contents, and Usage Data. Chapter 4.
- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Chapters 16 and 17.
- Ricardo Baeza-Yates and Berthier Ribeiro-Neto, Modern Information Retrieval, Chapter 2.
- Jure Leskovec, Anand Rajaraman, and Jeff Ullman, Mining of Massive Datasets, Chapters 7 and 11

Outline

Motivation

- **Problem:** Query terms can be ambiguous
  - E.g., query "star"retrieves documents about astronomy, animals, etc.

Motivation

- **Problem:** Query terms can be ambiguous
  - E.g., query "star"retrieves documents about astronomy, animals, etc.
  - Solution: clustering document responses to queries along lines of different topics

Motivation

- **Problem:** Query terms can be ambiguous
  - E.g., query "star"retrieves documents about astronomy, animals, etc.
  - Solution: clustering document responses to queries along lines of different topics
- **Problem:** Manual construction of topic hierarchies and taxonomies

Motivation

- **Problem:** Query terms can be ambiguous
  - E.g., query "star"retrieves documents about astronomy, animals, etc.
  - Solution: clustering document responses to queries along lines of different topics
- **Problem:** Manual construction of topic hierarchies and taxonomies
  - Solution: preliminary clustering of large samples of web documents

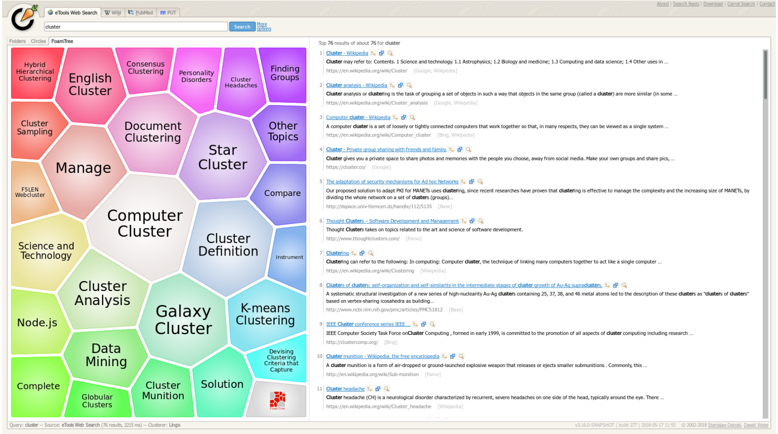
Motivation

- **Problem:** Query terms can be ambiguous
  - E.g., query "star"retrieves documents about astronomy, animals, etc.
  - Solution: clustering document responses to queries along lines of different topics
- **Problem:** Manual construction of topic hierarchies and taxonomies
  - Solution: preliminary clustering of large samples of web documents
- **Problem:** Speeding up similarity search

Motivation

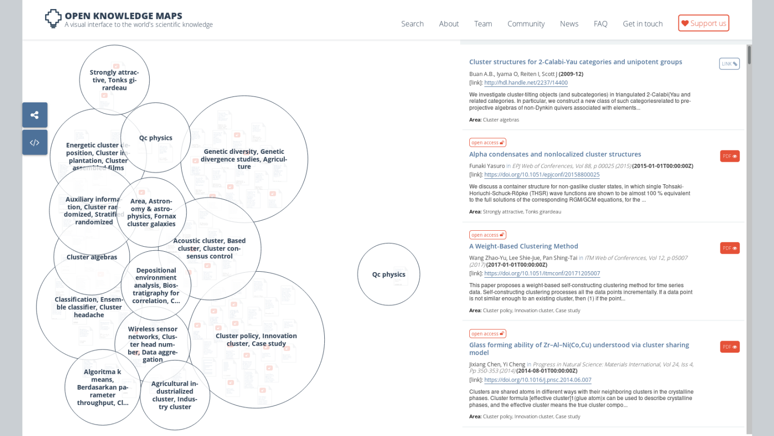
- **Problem:** Query terms can be ambiguous
  - E.g., query "star"retrieves documents about astronomy, animals, etc.
  - Solution: clustering document responses to queries along lines of different topics
- **Problem:** Manual construction of topic hierarchies and taxonomies
  - Solution: preliminary clustering of large samples of web documents
- **Problem:** Speeding up similarity search
  - Solution: restrict the search for documents similar to a query to most representative cluster(s)

An Example



http://search.carrot2.org

Another Example



https://openknowledgemaps.org/

Cluster Hypothesis:

- Given a 'suitable' clustering of a collection, if the user is interested in document d (or term t), he is likely to be interested in other members of the cluster to which d (t) belongs

Clustering Task:

- Use measures of similarity to cluster a collection of documents/terms into groups, so that similarity within a cluster is larger than across clusters

Clustering Concepts

- Clustering paradigms:
  - Bottom-up agglomerative clustering
  - Top-down partitioning
- Dimensionality reduction:
  - Embedding of corpus in a low-dimensional space
  - Many different approaches, based on heuristics, linear algebra, probabilistic models, ...

Clustering Approaches

- Partitioning Approaches
  - Bottom-up clustering
  - Top-down clustering
- Geometric Embedding Approaches
  - Self-organization map
  - Latent semantic indexing
- Generative models and probabilistic approaches
  - Single topic per document
  - Documents correspond to mixtures of multiple topics

Outline

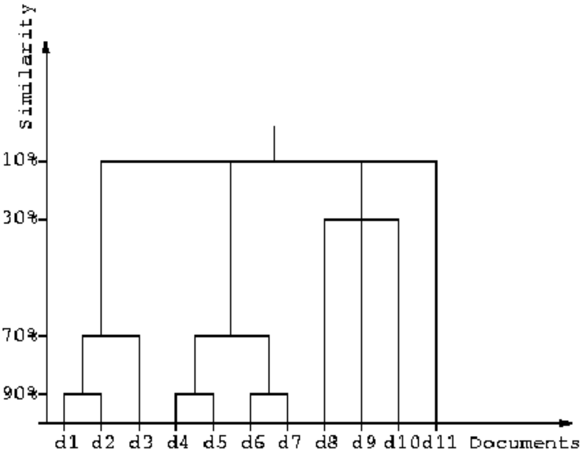
Partitioning Approaches

- Partition document collection into a set G of k clusters $[C_1, C_2, \ldots, C_k]$
- Choices:
  - Minimize intra-cluster distance: $\sum_i \sum_{d_1, d_2 \in C_i} \delta(d_1, d_2)$
  - Maximize intra-cluster semblance: $\sum_i \sum_{d_1, d_2 \in C_i} \rho(d_1, d_2)$
- If cluster representations $\vec{C}_i$ are available
  - Minimize $\sum_i \sum_{d \in C_i} \delta(d, \vec{C}_i)$
  - Maximize $\sum_i \sum_{d \in C_i} \rho(d, \vec{C}_i)$
- Soft clustering
  - d assigned to C_i with confidence $z_{d,i}$
  - Find $z_{d,i}$ so as to minimize $\sum_i \sum_{d \in C_i} z_{d,i}\delta(d, \vec{C}_i)$ or maximize $\sum_i \sum_{d \in C_i} z_{d,i}\rho(d, \vec{C}_i)$
- Two ways to get partitions: bottom-up clustering and top-down clustering

Hierarchical Agglomerative Clustering

- Consider that G represents a set of clusters
- Initially G is a collection of singleton groups, each with one document d
- Repeat
  - Find $\Gamma, \Delta \in G$ with max similarity measure, $s(\Gamma \cup \Delta)$
  - Merge group Γ with group Δ
- Use above info to plot the hierarchical merging process (dendogram)
- To get desired number of clusters: cut across any level of the dendogram

Example Dendogram



(b)

(example)

Similarity Measures

- Self-Similarity
  - Consider that Φ represents a cluster
  - Average pairwise similarity between documents in Φ

$$s(\Phi) = \frac{1}{\binom{|\Phi|}{2}} \sum_{d_1, d_2 \in \Phi} s(d_1, d_2)$$

  - $s(d_1, d_2)$ is the inter-document similarity measure (e.g., cosine of *TF-IDF* vectors)
- Other criteria:
  - Maximum/minimum pairwise similarity between documents in the clusters
- Complexity: $O(n^2 \log n)$ with n^2 space

Top-Down Clustering

- Use an internal representation for documents as well as clusters (centroids)
- Partition documents into k clusters
- 2 variants
  - Hard: 0/1 assignment of documents to clusters
  - Soft: documents belong to clusters with fractional scores
- Termination
  - When assignment of documents to clusters ceases to change much, or
  - when cluster centroids move negligibly over successive iterations

The k-Means Algorithm

Hard k-Means

- Choose k arbitrary centroids
- Assign each document to nearest centroid
- Recompute centroids

- Contribution for updating cluster centroid

$$\Delta\mu_c = \sum_d \begin{cases} \eta(d - \mu_c) & \text{if } \mu_c \text{ is closest to } d \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_c \leftarrow \mu_c + \Delta\mu_c$$

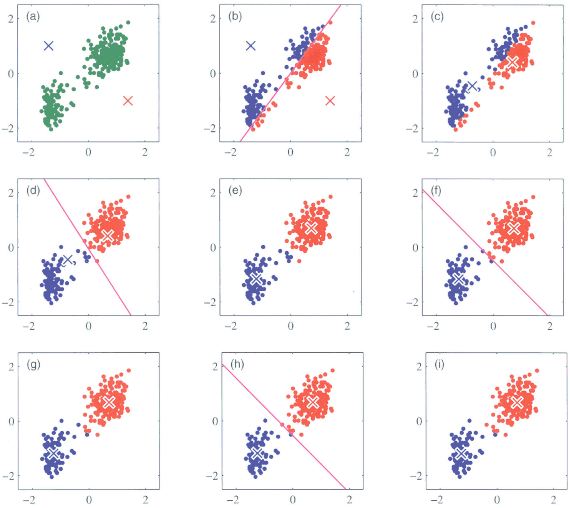
The k-Means Algorithm

Soft k-Means

- Don't break close ties between document assignments to clusters and don't make documents contribute to a single cluster which wins narrowly
  - The contribution for updating cluster centroid μ_c from document d is related to the current similarity between both

$$\Delta \mu_c = \eta \frac{1/|d - \mu_c|^2}{\sum_\gamma 1/|d - \mu_\gamma|^2}(d - \mu_c)$$

An Example



(online demo)

Outline

Dimensionality Reduction

- **Goal:** Embedding of corpus in a low-dimensional space
- Self-Organizing Map (SOM)
  - Technique related to *k*-means
- Latent Semantic Indexing (LSI)
  - Linear transformations to reduce number of dimensions
- Other techniques:
  - Multidimensional scaling (MDS): Minimize the distortion of interpoint distances in the low-dimensional embedding as compared to the dissimilarity given in the input data
  - NN-based word embeddings: Use neural networks to learn a mapping from the high dimensional vocabulary space to a lower dimension concept-based space

Self-Organizing Maps

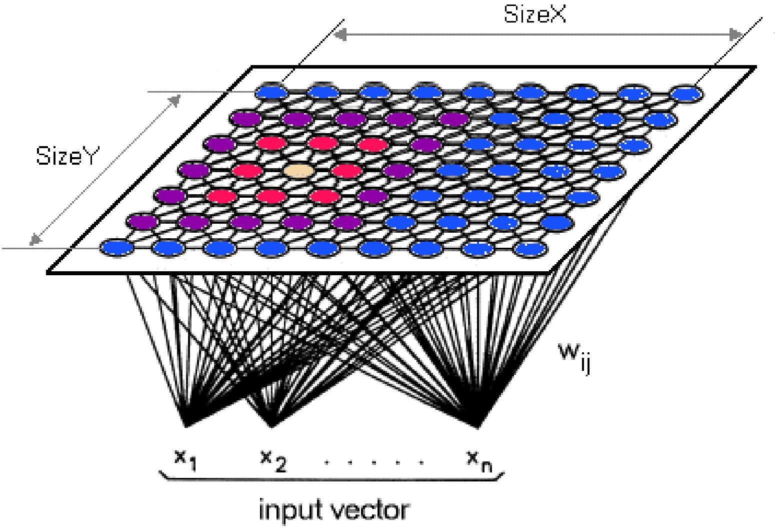
- Like soft k-means
  - Determine association between clusters and documents
  - Associate a representative vector with each cluster and iteratively refine
- Unlike k-means
  - Embed the clusters in a low-dimensional space right from the beginning
  - A large number of clusters can be initialized even if eventually many are to remain devoid of documents
- Each cluster can be a slot in a square/hexagonal grid.
  - The grid structure defines the neighborhood $N(c)$ for each cluster c
- Also involves a proximity function $h(\gamma, c)$ between clusters

Update Rule

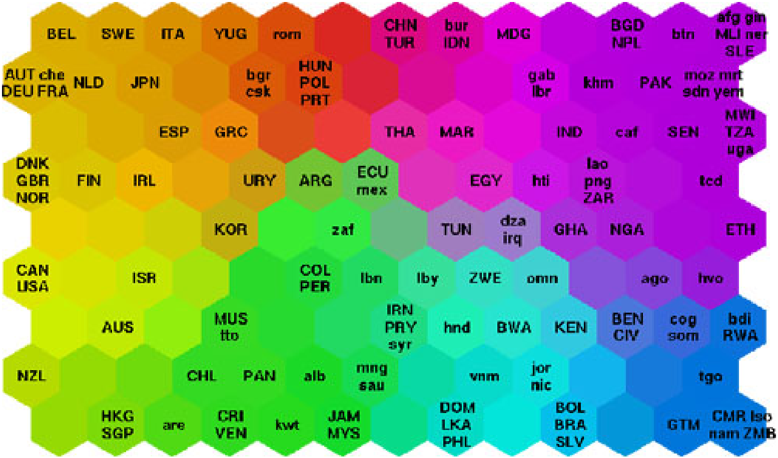
- Data item d activates node (closest cluster) as well as the neighborhood nodes
  - E.g., all nodes within n hops
- Update rule for node under the influence of d is:

$$\mu_\gamma \leftarrow \mu_\gamma + \eta h(\gamma, c_d)(d - \mu_\gamma)$$

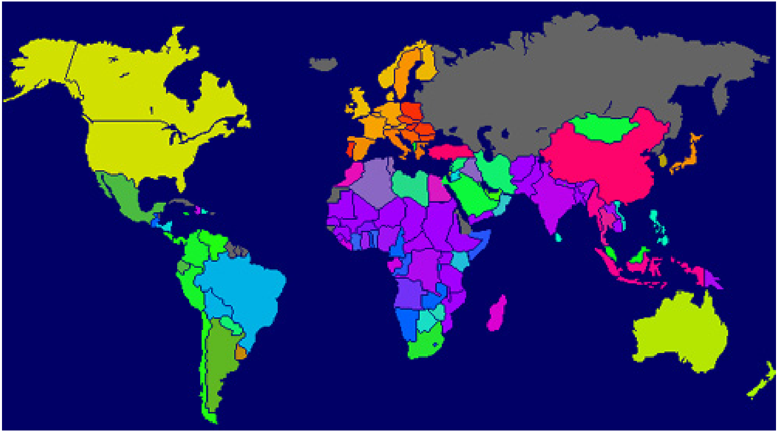
A Self Organizing Map



Example


BEL
SWE
ITA
YUG
rom
CHN TUR
bur IDN
MDG
BGD NPL
btn
afg gin MLI ner SLE
AUT che DEU FRA
NLD
JPN
bgr csk
HUN POL PRT
gab lbr
khm
PAK
moz mrt sdn yem
ESP
GRC
THA
MAR
IND
caf
SEN
MWI TZA uga
DNK GBR NOR
FIN
IRL
URY
ARG
ECU mex
EGY
hti
lao png ZAR
tcd
KOR
zaf
TUN
dza irq
GHA
NGA
ETH
CAN USA
ISR
COL PER
lbn
lby
ZWE
omn
ago
hvo
AUS
MUS tto
IRN PRY syr
hnd
BWA
KEN
BEN CIV
cog som
bdi RWA
NZL
CHL
PAN
alb
mng sau
vnm
jor nic
tgo
HKG SGP
are
CRI VEN
kwt
JAM MYS
DOM LKA PHL
BOL BRA SLV
GTM
CMR lso nam ZMB

Example



(online demo)

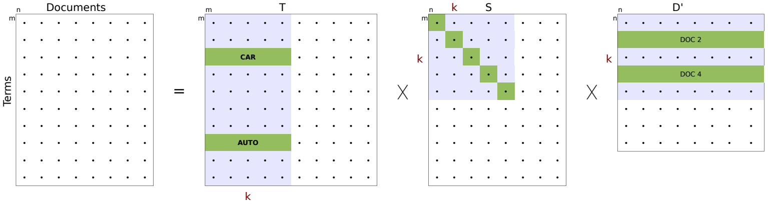
Extended Similarity and LSI

- In the vector space model different terms have different meanings
  - car $\neq$ automobile
- However, "car" and "automobile" are related
  - They are likely to co-occur often
- Documents having related words are related
  - Useful for search and clustering
- Two basic approaches:
  - Hand-made thesaurus (e.g., WordNet)
  - Co-occurrence and associations

Latent Semantic Indexing

- Vector-space model
  - Distinct orthogonal direction for each term
- Not all terms are orthogonal
  - E.g., "car" and "automobile"
- We need a matrix with a lower rank than the traditional document-term matrix, where similar terms are "merged"
- This matrix can be found through Singular Value Decomposition

Singular Value Decomposition



Querying

- $A = T \times S \times D^T \Leftrightarrow D = A^T \times T \times S^{-1}$
- A query q is also projected to the new space

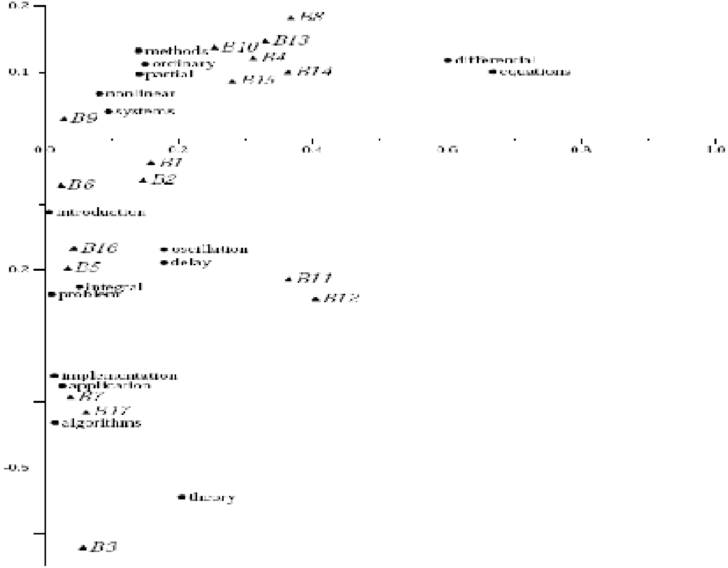
$$\hat{q} = q^T \times T \times S^{-1}$$

- Cosine similarity can now be applied to the projections
- Results are often better than standard *TF-IDF* retrieval/classification
  - SVD filters out noise and "discovers" semantic associations between terms
  - Representations in the projected vector space can be used for visualization

Example

Label	Titles
B1	A Course on Integral Equations
B2	Attractors for Semigroups and Evolution Equations
B3	Automatic Differentiation of Algorithms: Theory, Implementation, and Application
B4	Geometrical Aspects of Partial Differential Equations
B5	Ideals, Varieties, and Algorithms An Introduction to Computational Algebraic Geometry and Commutative Algebra
B6	Introduction to Hamiltonian Dynamical Systems and the N-Body Problem
B7	Knapsack Problems: Algorithms and Computer Implementations
B8	Methods of Solving Singular Systems of Ordinary Differential Equations
B9	Nonlinear Systems
B10	Ordinary Differential Equations
B11	Oscillation Theory for Neutral Differential Equations with Delay
B12	Oscillation Theory of Delay Differential Equations
B13	Pseudodifferential Operators and Nonlinear Partial Differential Equations
B14	Sinc Methods for Quadrature and Differential Equations
B15	Stability of Stochastic Differential Equations with Respect to Semi-Martingales
B16	The Boundary Integral Approach to Static and Dynamic Contact Problems
B17	The Double Mellin-Barnes Type Integrals and Their Applications to Convolution Theory

Example


B8
B13
methods
B10
B4
ordinary
partial
B14
B15
differential
equations
nonlinear
systems
B9
B1
B2
B6
introduction
B16
oscillation
delay
B5
B11
integral
problem
B12
implementation
application
B7
B17
algorithms
theory
B3

Problems

- Exact solution is computationally expensive
  - Can run in minutes to hours on a 10^3 to 10^4 collection
- Approximation algorithms frequently used in practice
  - Implementation in *scikit-learn* uses a fast randomized SVD solver [Halko , 2009]
- Most current implementations need to store the whole input matrix in memory
- Still not feasible for Web-sized collections

Questions?