Information Retrieval

WS 2016 / 2017

Lecture 9, Tuesday December 20th, 2016

(Clustering, K-Means)

Prof. Dr. Hannah Bast
Chair of Algorithms and Data Structures
Department of Computer Science
University of Freiburg

UNI FREIBURG

Overview of this lecture

- Organizational
  - Your experiences with ES8 — Vector Space Model
  - Christmas present — No lectures for two weeks
- Contents
  - Clustering — Definition and example
  - K-Means — Algorithm and analysis
  - K-Means for text — Implementation advice

ES9: cluster our movies dataset using k-means, then report run-time and cluster quality on the Wiki

Experiences with ES8 1/5

UNI FREIBURG

- Summary / excerpts
  - Conceptually easy + good introduction to linear algebra
  - More time than expected, first-time use of Python for some
  - Getting used to numpy and scipy cost some time

    Will be useful for the next three lectures as well !
  - "It was a bit of a kerfuffle coding everything"

    active_vocabulary_size++ … thanks!
  - Multiplication of dense and sparse matrix → out of memory
  - What is more real, consciousness or matter?

    You quoted: Descartes, Frida Kahlo, The Matrix, Harry Potter

Experiences with ES8 2/5

- Results
  - Three score variants : tf, tf.idf, BM25
  - Two normalization variants : normalized or not
  - Note that normalization does a similar thing as the document length normalization of BM25
  - Correspondingly, the results had the following tendency:

    Better results for tf and tf.idf with normalization (slightly)

    Worse results for BM25 with normalization (slightly)

    The unnormalized BM25 scores gave the best results

Experiences with ES8 3/5

- What is more real: consciousness or matter?
  - Arguments against "matter is real":

    Recall from L8: our view of the world is extremely selective, conceptual, and biased

    An example: in Australia, the jewel beetle prefers copulation with a special kind of brown beer bottle, because it looks like a gigantic female to it → species almost driven to extinction

    The situation is slightly better for humans

    But in principle, we have exactly the same problem … and like the beetle we might not have the necessary means to realize that ourselves

Experiences with ES8 4/5

- What is more real: consciousness or matter?
  - Arguments for "matter is real":

    Using our capability for abstract thinking, we have developed a complex model of matter that can predict even complex and unusual phenomena with great precision

    For example, starlight bending around massive objects

    It therefore seems that matter exists by itself (also without a consciousness observing it) since different people can measure the same thing and get the same result

    However, that might not be true … maybe we are all actually part of one consciousness (without realizing it) and experiencing the same collective illusion

Experiences with ES8 5/5

- What is more real: consciousness or matter?
  - Arguments for "consciousness is more real":

    Even if we have no idea what consciousness is and how it works, it is obvious that is exists, because we are experiencing it live right now

    That is what Descartes meant by his "Cogito ergo sum"

    Everything else could, in principle, be just an elaborate illusion created in our consciousness (in whatever way), and which we perceive as the world around us

    It is impossible for us to know that for sure, all the arguments from the previous slide are just "indications"

Clustering 1/3

- Informal definition
  - Given elements $x_1, \ldots, x_n$ from a **metric space**

    metric space = there is a measure of distance between any two elements
  - Group the elements into clusters $C_1, \ldots, C_k$ such that

    Intra-cluster distances are as small as possible

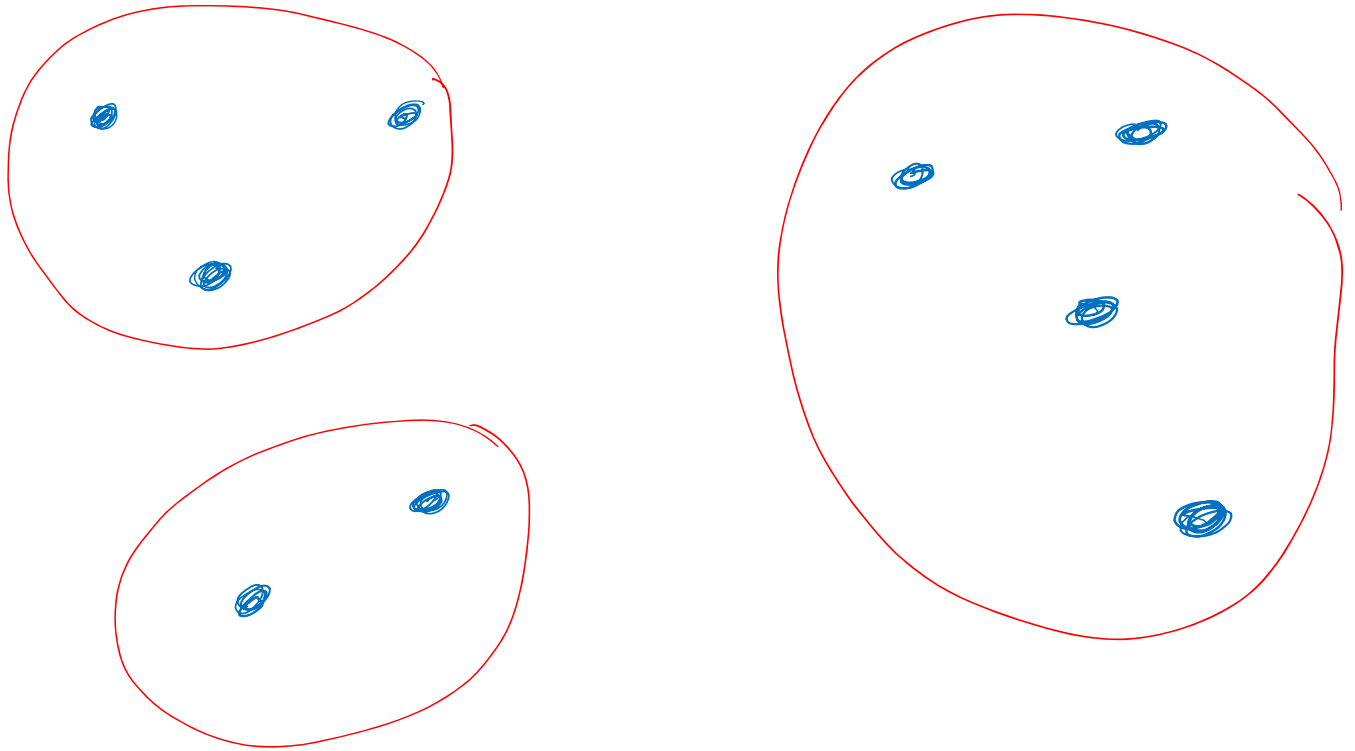
    Inter-cluster distances are as large as possible

    We will make this more precise on slide 10
  - We assume that k is given as part of the input

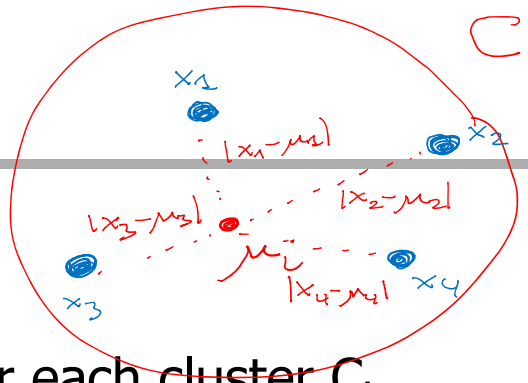
Clustering 2/3

$k = 3$

- Example



Clustering 3/3



- Centroids and RSS
  - Assume we have a **centroid** μ_i for each cluster C_i

    Intuitively: a single element from the metric space "representing the cluster"
  - Let c_i be the index of the cluster to which x_i is assigned

    Each element belongs to <u>one</u> cluster = **hard** clustering
  - Then we define the **residual sum of squares** as

    $RSS = \Sigma_{i=1,\ldots,k} \Sigma_{x \in C_i} (x - \mu_i)^2 = \Sigma_{i=1,\ldots,n} (x_i - \mu_{c_i})^2$

    The sum of the squares of all intra-cluster distances

K-Means 1/9

- Algorithm
  - Basic idea: find a local optimum of the RSS by greedily minimizing it in every step
  - Initialization: pick a set of centroids

    For ES9, pick k random documents from the input set
  - Then alternate between the following two steps

    (A) Assign each element to its nearest centroid

    (B) compute new centroids as average of elems assigned to it
  - Let's first look at a nice demo and then show that both steps can only **decrease** the RSS

K-Means 2/9

- Step A (assign to nearest centroid)
  - Recall: $RSS = \Sigma_{i=1,\ldots,n} (x_i - \mu_{ci})^2$
  - In Step A, the centroids $\mu_1, \ldots, \mu_k$ are fixed and we want to find those $c_1, \ldots, c_n$ that minimize the RSS:

    $\min_{c1,\ldots,cn} \Sigma_{i=1,\ldots,n} (x_i - \mu_{ci})^2 = \Sigma_{i=1,\ldots,n} \min_{ci} (x_i - \mu_{ci})^2$

    Each summand can be minimized independently
  - $\min_{ci} (x_i - \mu_{ci})^2 = \min_{ci} |x_i - \mu_{ci}|$

    The square distance is min. when the distance is min.
  - $|x_i - \mu_{ci}|$ is minimized for $c_i = \text{argmin}_j |x_i - \mu_j|$

    In words: by assigning x_i to its nearest centroid

K-Means 3/9

- Step B (recompute centroids)
  - Recall: $RSS = \Sigma_{i=1,\ldots,k} \Sigma_{x \in Ci} (x - \mu_i)^2$
  - In Step B, the clusters $C_1, \ldots, C_k$ are fixed and we want to find the centroids $\mu_1, \ldots, \mu_k$ that minimize the RSS:

    $\min_{\mu 1,\ldots,\mu n} \Sigma_{i=1,\ldots,k} \Sigma_{x \in Ci} (x - \mu_i)^2 = \Sigma_{i=1,\ldots,k} \min_{\mu i} \Sigma_{x \in Ci} (x - \mu_i)^2$

    The RSS part for each cluster can be minimized independently
  - We can solve $\min_{\mu i} \Sigma_{x \in Ci} (x - \mu_i)^2$ using simple calculus:

$$\frac{\partial}{\partial \mu_i} \sum_{x \in C_i} (x - \mu_i)^2 = -2 \sum_{x \in C_i} (x - \mu_i) \stackrel{!}{=} 0$$

$$\Rightarrow \sum_{x \in C_i} x = \sum_{x \in C_i} \mu_i = |C_i| \cdot \mu_i \Rightarrow \mu_i = \sum_{x \in C_i} x / |C_i| \quad \square$$

$$\frac{\partial^2}{\partial \mu_i^2} \sum_{x \in C_i} (x - \mu_i)^2 = 2|C_i| > 0 \Rightarrow \text{Minimum!}$$

K-Means 4/9

n = #elements
k = # clusters
clusterings = $\underbrace{k \cdot \ldots \cdot k}_{n \text{ times}} = k^n$

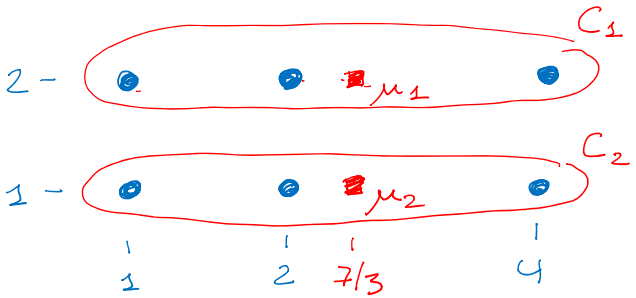
- Convergence to local RSS minimum
  - By what we have just proven, RSS stays equal or decreases in every step (A) and every step (B)
  - There are only finitely many clusterings
  - Therefore, the algorithm will terminate if we avoid that it cycles forever between different clusterings with equal RSS
  - Solution: deterministic tie breaking in the centroid assignment, when two centroids are equally close

    For ES9, simply prefer the centroid with smaller index

    Not needed for last termination condition from slide 17

K-Means 5/9

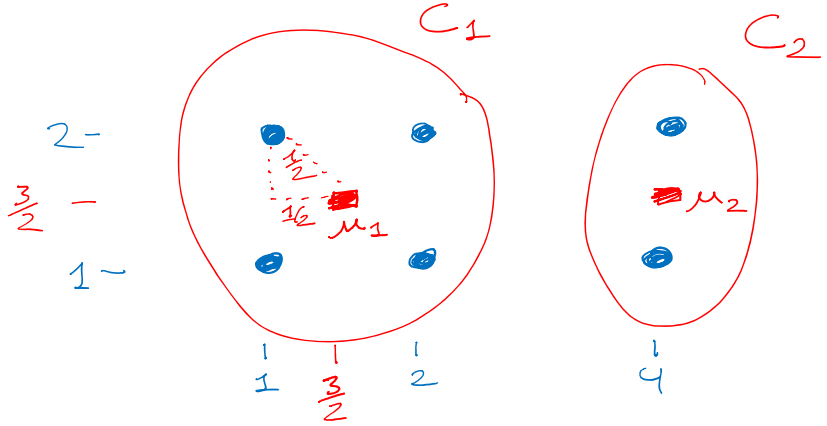
- A local RSS minimum is not always a global one



$$RSS = 2 \cdot \left(\left(1 - \tfrac{7}{3}\right)^2 + \left(2 - \tfrac{7}{3}\right)^2 + \left(4 - \tfrac{7}{3}\right)^2 \right)$$

$$= 2 \cdot \left(\tfrac{16}{9} + \tfrac{1}{9} + \tfrac{25}{9} \right)$$

$$= \tfrac{84}{9} = 9\tfrac{1}{3}$$



$$RSS = 4 \cdot \left(\tfrac{1}{2^2} + \tfrac{1}{2^2} \right) + 2 \cdot \tfrac{1}{2^2}$$

$$= 4 \cdot \tfrac{1}{2} + \tfrac{1}{2}$$

$$= 2\tfrac{1}{2}$$

much better !

K-Means 6/9

- Termination condition, options
  - **Stop** when no more change in clustering

    Optimal, but this can take a **very** long time
  - **Stop** after a fixed number of iterations

    Easy, but how to guess the right number?
  - **Stop** when RSS falls below a given threshold

    Reasonable, but RSS may never fall below that threshold
  - **Stop** when decrease in RSS falls below a given threshold

    Reasonable: we stop when we are close to convergence

    For ES9, aim at a combination of small final RSS and a fast running time … post results on the Wiki

K-Means 7/9

- Choice of a good k
  - **Idea 1:** choose the k with smallest RSS

    Bad idea, because RSS is minimized for k = n
  - **Idea 2:** choose the k with smallest RSS + λ · k

    Makes sense: RSS becomes smaller as k becomes larger

    But now we have λ as a tuning parameter

    Experience shows that for a given kind of application, there is often an input-independent good choice for λ, whereas a good k depends on the input

    For ES9, the number of clusters is given

K-Means 8/9

- When is K-Means a good clustering algorithm
  - K-Means tends to produce compact clusters of about equal size

    Indeed, it can be shown that K-Means is optimal when the sought for clusters are spherical and of equal size

    Whether it's good or not, k-means is used a **lot lot lot** in practice, just because of it's simplicity

K-Means 9/9

- Alternatives
  - **K-Medoids**

    Maintain that centroids are elements from the input set
  - **Fuzzy k-means**

    Elements can belong to several clusters to varying degrees ... this is sometimes called "soft clustering"

    L10 will be about a method for soft clustering (LSI)
  - **EM-Algorithm** (EM = Expectation-Maximization)

    General-purpose optimization technique that can also be used for soft clustering

K-Means for Text Documents 1/7

- Representation
  - We use the vector space model (VSM), as in Lecture 8

    Each document = one column of our term-doc matrix
  - Centroids are also vectors in this space
  - To computer the centroid of a set of documents, just take the average of the document vectors
  - Important observation: the document vectors are **sparse**, the centroids become **dense** over time

    For ES9, it is critical that you store the document vectors in sparse representation, for the same reasons as in ES8

K-Means for Text Documents 2/7

- Construct from an inverted index
  - The term-document matrix can be constructed from an inverted index just as shown in the last lecture

    For ES9, you can re-use your code from ES8, or from the master solutions if you prefer

    You need to write very little additional code, but clever code = using the right linear algebra operations

    Figuring out these operations is interesting and fun

K-Means for Text Documents 3/7

- Running time
  - Let n = #documents, m = #terms, k = #clusters
  - Assume that each dist computation takes time Θ(D)
  - Then each step (A) takes time **Θ(k · n · D)**

    Compute **dist** between each documents and each cluster
  - Each step (B) takes time **Θ(n · m)**

    Each of the n documents is added to one centroid vector, and one vector addition takes time Θ(m)

D = # non-zero entries in the two vectors

$|x - y| = \sum_{i=1}^{m} (x_i - y_i)^2$

K-Means for Text Documents 4/7

- Distance between two documents
  - We use Euclidean distance between the normalized docs:
    dist(x, y) := |x' – y'|, where x' = x / |x| and y' = y / |y|

    Straightforward computation between sparse and dense vector takes time Θ(m), where m = total number of terms
  - **Lemma:** $|x - y|^2 = |x|^2 + |y|^2 - 2 \cdot x \bullet y$, where $x \bullet y$ is the dot product of x and y

    Hence: when |x| = |y| = 1, then $\frac{1}{2} \cdot |x - y|^2 = 1 - x \bullet y$

    Computing the dot product for a sparse x and a dense y takes time Θ(M), where M = number of non-zero entries in x

    Note: when |x| = |y| = 1, then $x \bullet y$ is the cosine of the angle between x and y … a common similarity measure in IR

K-Means for Text Documents 5/7

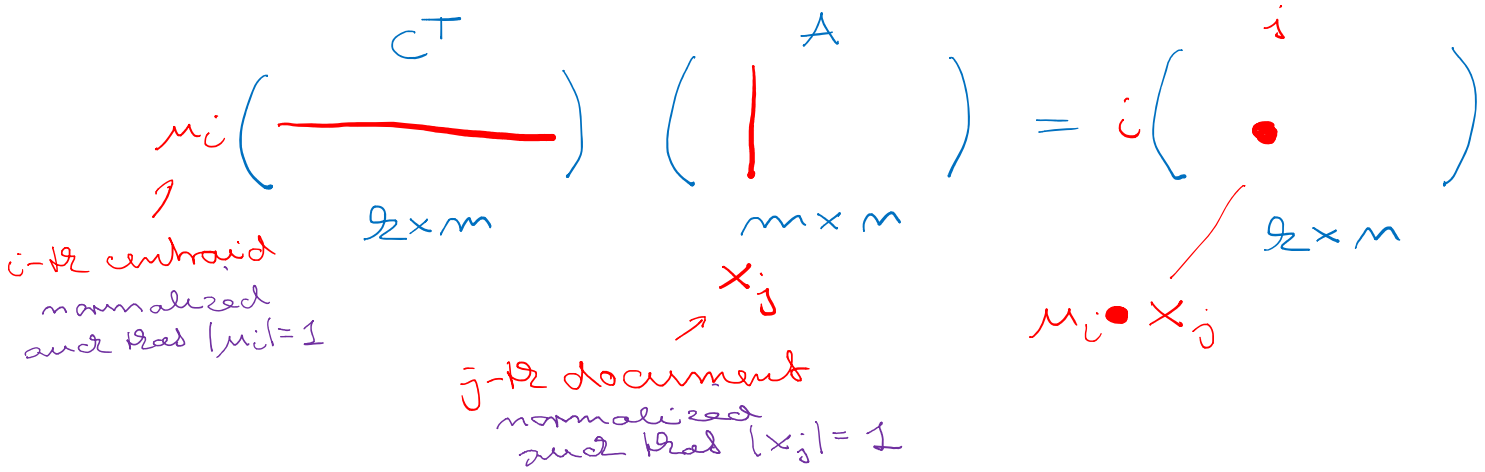
- Using matrix operations
  - Both Steps (A) and (B) can be implemented very efficiently using matrix operations

    Some hints and examples on the next two slides
  - Use the lemma from the previous slides and make sure that document vectors and centroids are L_2-normalized

    You can reuse the L_2-normalization code from your solution to ES8 or from the master solution

K-Means for Text Documents 6/7

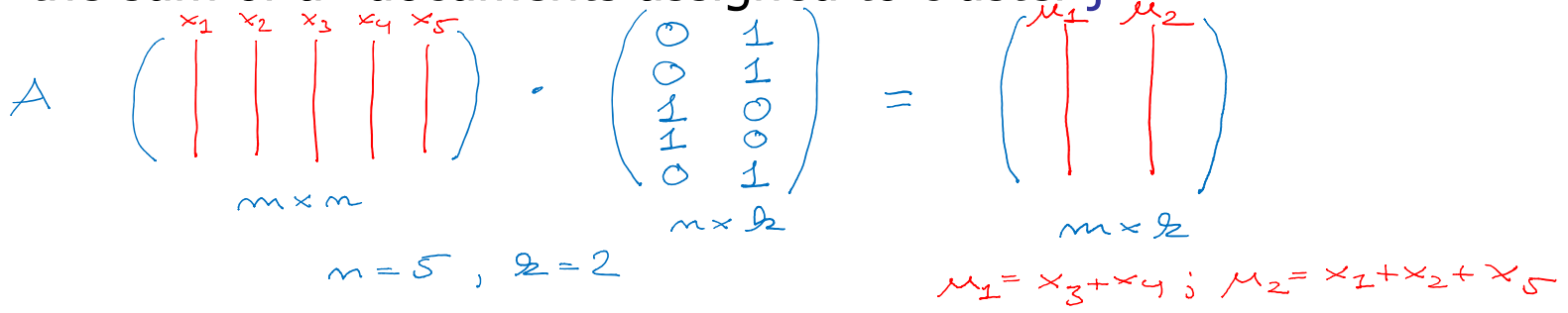
- Using matrix operations, Step (A)
  - For Step (A), we need to compute the dot products between all documents and all centroids
  - Let A be the term-document matrix (one doc per column)
  - Let C be the term-centroid matrix (one centroid per column)
  - Then $C^T \cdot A$ yields a matrix, where the entry at i, j is exactly the dot product between centroid i and document j



K-Means for Text Documents 7/7

- Using matrix operations, Step (B)
  - For Step (B), we need to **add** the vectors of all documents in the same cluster C, and then divide by |C|

    Since we normalize afterwards, we can drop "divide by |C|"
  - Let A be the term-document matrix (one doc per column)
  - Let B be a 0-1 matrix where the entry at i, j is 1 iff document i is in cluster j
  - Then A · B yields a matrix, where the j-th column is exactly the sum of all documents assigned to cluster j



References

- Further reading
  - Textbook Chapter 16: Flat clustering

    http://nlp.stanford.edu/IR-book/pdf/16flat.pdf
- Wikipedia
  - http://en.wikipedia.org/wiki/Cluster_analysis
  - http://en.wikipedia.org/wiki/K-means
  - http://en.wikipedia.org/wiki/K-medoids
  - http://en.wikipedia.org/wiki/EM_Algorithm