Information Retrieval

WS 2017 / 2018

Lecture 9, Tuesday December 19th, 2017

(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 break, 2 weeks — U+1F600 U+1F36A
- Contents
  - Clustering — Definition and example
  - K-Means — Algorithm and analysis
  - K-Means for text — All kinds of practical advice

ES9: a few pencil-and-paper exercises which should test and deepen your understanding

(this is the kind of tasks you might also get in the exam)

Experiences with ES8 1/2

- Summary / excerpts
  - Conceptually easy, but the devil is in the details

    "I like this sheet: not too hard, but one needs to be careful"
  - Tricky to figure out how to do things efficiently in scipy

    "realized quickly that todense method is evil"
  - At least one of you used Java and regretted it (linear algebra code is so much more convenient in Python)
  - Many of you still catching up on your programming practice

    "I feel like I am learning more and more about Python and Java programming … thank you, I realized that I really need more training … I am having fun (I guess)."

Experiences with ES8 2/2

- Results
  - Three score variants : BM25 and tf
  - Three normalizations : rows, columns, none
  - Understand that:

    Row normalization is similar (not identical) to idf

    Column normalization is similar (not identical) to the document length normalization of BM25
  - The best results were achieved with BM25

    Which is more evidence (based on very small benchmark though) that the BM25 normalizations are indeed good

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 from the space
  - Group the elements into clusters $C_1, \ldots, C_k$ such that

    Intra-cluster distances are as small as possible

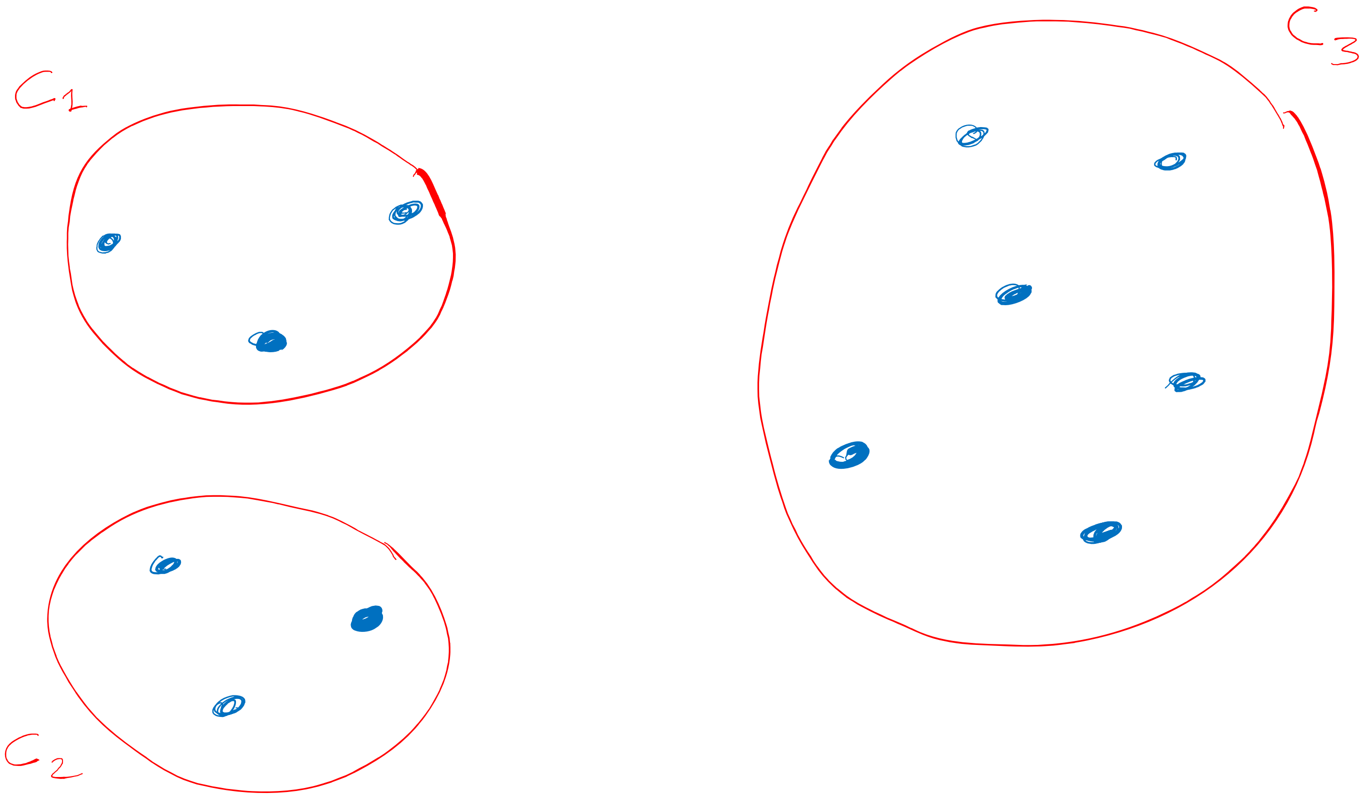
    Inter-cluster distances are as large as possible

    We will make this notion more precise on slide 7
  - We assume that k (the number of clusters) is given as part of the input

Clustering 2/3

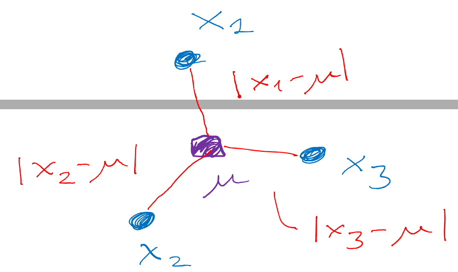
$k = 3$

- Example



Clustering 3/3

one cluster



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

    Intuitively, a centroid is a single element from the metric space that "represents" 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, in Lecture 10 we will also see soft clustering
  - Then we define the **residual sum of squares** as

    $(x_i - \mu)^2 = |x_i - \mu|^2$

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

    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 example, pick a random subset from the input
  - 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: $\text{RSS} = \Sigma_{i=1,\ldots,k} \Sigma_{x \in C_i} (x - \mu_i)^2$
  - 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:

$$\min_{\mu_1,\ldots,\mu_n} \overbrace{\Sigma_{i=1,\ldots,k} \Sigma_{x \in C_i} (x - \mu_i)^2}^{\text{RSS}} = \Sigma_{i=1,\ldots,k} \min_{\mu_i} \Sigma_{x \in C_i} (x - \mu_i)^2$$

  - Each $\min_{\mu_i} \Sigma_{x \in C_i} (x - \mu_i)^2$ can be solved independently, and we can do that using simple calculus:

concave function

$$f(\mu) = \Sigma_{x \in C} (x - \mu)^2$$

$$f'(\mu^*) = -2 \Sigma_{x \in C} (x - \mu^*) \overset{!}{=} 0$$

$$\Leftrightarrow \sum_{x \in C} x - \underbrace{\sum_{x \in C} \mu^*}_{= |C| \cdot \mu} = 0 \Leftrightarrow \mu^* = \frac{1}{|C|} \cdot \sum_{x \in C} x$$

$$f''(\mu) = 2 \sum_{x \in C} 1 = 2|C| > 0$$

$$\Rightarrow \text{MIN at } \mu^*$$

K-Means 4/9

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

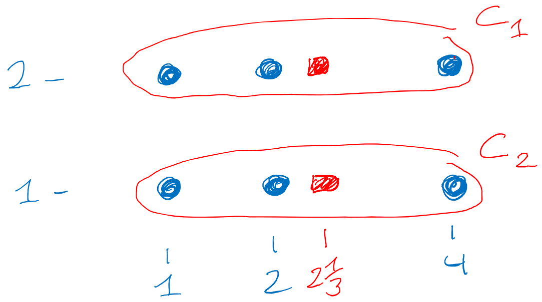
- Convergence to local RSS minimum
  - By what we have just proven, RSS stays equal or decreases in every step (A) and every step (B)
  - There are only finitely many clusterings ... think: how many?
  - Therefore, the algorithm will terminate if we avoid that it cycles forever between different clusterings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

    For ES9 prefer the centroid with larger index

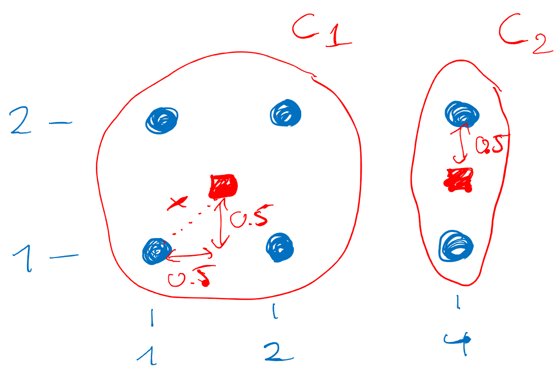
    Some termination conditions don't need a tie-breaking rule, for example, the last condition from slide 13

K-Means 5/9

- A local RSS minimum is not always a global one



$$RSS = 2 \cdot \left(\left(\frac{4}{3}\right)^2 + \left(\frac{1}{3}\right)^2 + \left(\frac{5}{3}\right)^2 \right)$$
$$= 2 \cdot \frac{16 + 1 + 25}{9}$$
$$= 2 \cdot \frac{42}{9} = \frac{84}{9} = 9\frac{1}{3}$$



$$x^2 = 0.5^2 + 0.5^2 = 0.5$$

$$RSS = 4 \cdot 0.5 + 2 \cdot 0.25$$
$$= 2.5$$

Quite a bit smaller!

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 the toy example of ES9.2 take the first condition

K-Means 7/9

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

    then RSS = 0

    Bad idea, because RSS is minimized for k = n
  - **Idea 2:** choose the k with smallest RSS + $\lambda \cdot k$

    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 always 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 1/6

- Representation
  - We use the vector space model (VSM) from 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

    When implementing k-means, document vectors must be stored in sparse representation, just like in L8
  - Note: the term-document matrix can be constructed from an inverted index just as shown in the last lecture

K-Means for Text Documents 2/6

- Running time
  - Let n = #documents, m = #terms, k = #clusters
  - Assume that each dist computation takes time $\Theta(D)$
  - Then each step (A) takes time $\boldsymbol{\Theta(k \cdot n \cdot D)}$

    Compute **dist** between each documents and each cluster
  - Each step (B) takes time $\boldsymbol{\Theta(n \cdot m)}$

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

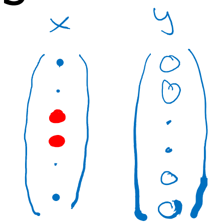
K-Means for Text Documents 3/6

$|x - y| = \sqrt{\sum_i (x_i - y_i)^2}$

PROOF OF LEMMA:

$|x - y|^2 = (x - y)^2$
$= (x - y) \bullet (x - y)$
$= \underbrace{x \bullet x}_{|x|^2} + \underbrace{y \bullet y}_{|y|^2} - 2 \cdot x \bullet y$

- 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 $\frac{1}{2} \cdot |x - y|^2 = 1 - x \bullet y$

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

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

K-Means for Text Documents 4/6

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

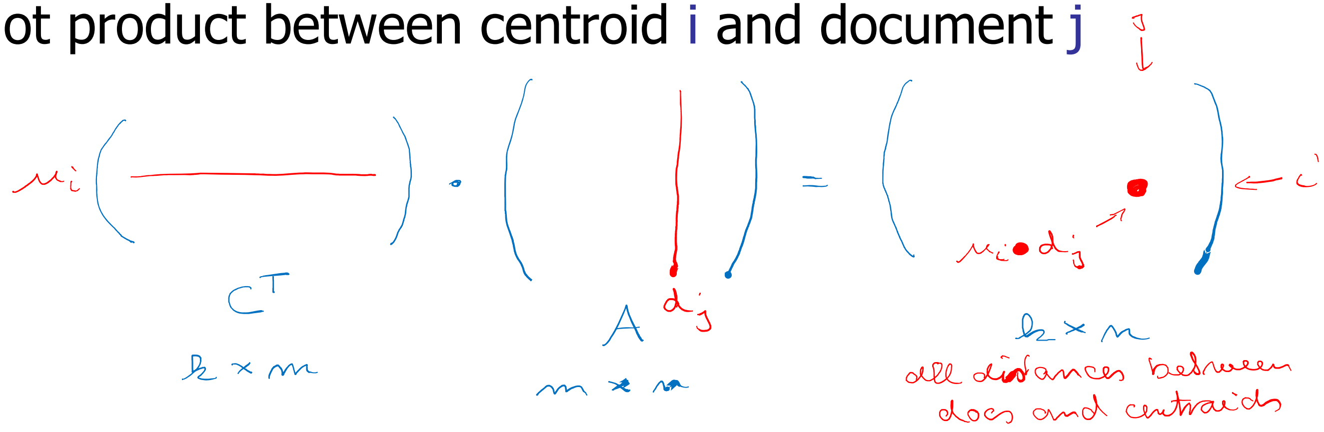
    Some hints and examples on the next two slides
  - Use the lemma from the previous slides and make sure that the centroids are L_2-normalized
  - Understand that documents do **not** have to be normalized:

    For each document, all we want to know is which centroid is the closest; multiplying that document by a constant factor does not change the relative distances to the centroids

    Important take-home message for life: understanding the underlying mathematics can save you **a lot** of work

K-Means for Text Documents 5/6

- 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 6/6

- 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|

    If one normalizes afterwards, one 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

$n = 5$

$k = 2$

$$\underbrace{\begin{array}{ccccc} d_1 & d_2 & d_3 & d_4 & d_5 \\ \end{array}}_{} \qquad$$

$$\underset{m \times 5}{\underset{A}{\begin{pmatrix} | & | & | & | & | \end{pmatrix}}} \cdot \underset{5 \times 2}{\underset{B}{\begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}}} = \underset{m \times 2}{\begin{pmatrix} | & | \\ \mu_1 & \mu_2 \end{pmatrix}}$$

$\mu_1 = d_3 + d_4$

$\mu_2 = d_1 + d_2 + d_5$

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