0007 sheet 1 učo points

Consider the following collection of four documents d_i:

- d_1: BREAKTHROUGH DRUG FOR HIV
- d_2: NEW HIV DRUG
- d_3: NEW APPROACH FOR TREATMENT OF HIV
- d_4: NEW HOPES FOR HIV PATIENTS

Produce a list of (term, document ID) tuples [1 point], sort this list in lexicographical order [1 point], and use the sorted list to construct an inverted index [1 point]. Write down each step. Describe how you would produce this index using the MapReduce distributed framework [2 points].

(breakthrough, 1), (drug, 1), (for, 1), (HIV, 1),
(new, 2), (HIV, 2), (drug, 2),
(new, 3), (approach, 3), (for, 3), (treatment, 3), (of, 3), (HIV, 3),
(new, 4), (hopes, 4), (for, 4), (HIV, 4), (patients, 4)

(approach, 3), (breakthrough, 1), (drug, 1), (drug, 2), (for, 1),
(for, 3), (for, 4), (HIV, 1), (HIV, 2), (HIV, 3), (HIV, 4),
(hopes, 4), (new, 2), (new, 3), (new, 4), (of, 3), (patients, 4),
(treatment, 3)

approach → 3
breakthrough → 1
drug → 1 → 2
for → 1 → 3 → 4
HIV → 1 → 2 → 3 → 4
hopes → 4
new → 2 → 3 → 4
of → 3
patients → 4
treatment → 3

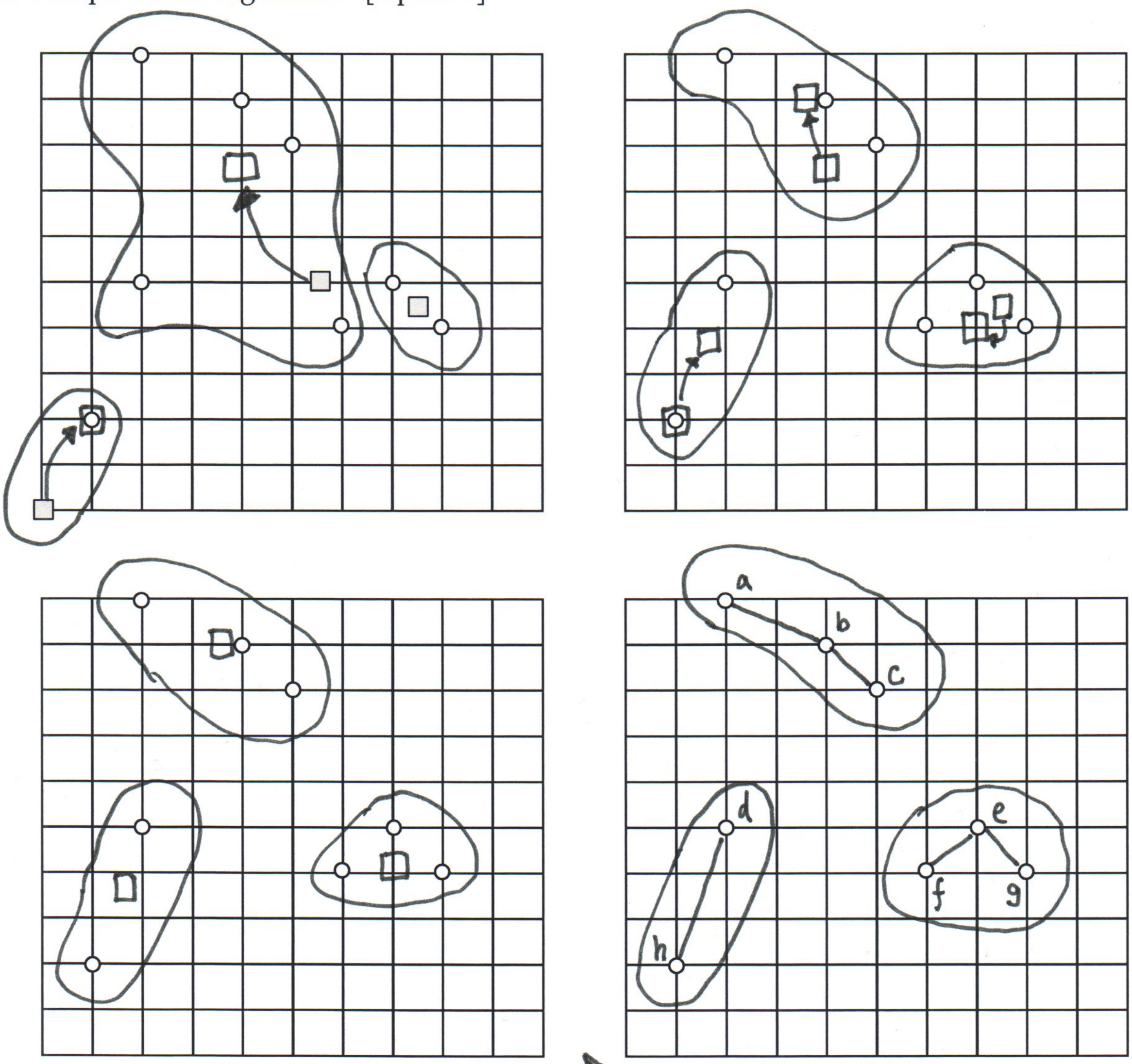
Each parser would process a limited number of documents and produce a sorted (term, document ID) list.
Each inverter would take all sublists in a certain alphabetical range of terms and produce postings for that alphabetical range.

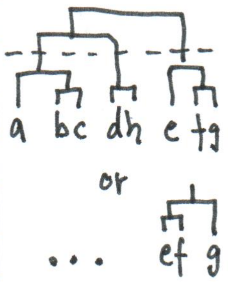
Explain the following aspects of the K-means flat clustering algorithm [2 points]:

1. What do we need to know about our dataset before using the algorithm?
2. What is the input and the output of the algorithm?
3. What are the two steps that take place in every epoch?
4. How do we decide in which epoch to stop the algorithm?

1. The number of classes K and initial mean estimates (seeds).
2. I: unclassified points and K seeds. O: K groups (clusters) of points.
3. Reassigning points, recomputing centroids.
4. Centroids converged.

Given the points ○, and the seeds □, run the K-means algorithm for three epochs. Draw the state of the algorithm at the beginning and after every epoch; no computation should be necessary. What is the output of the algorithm? [2 points]





Perform a hierarchical clustering of the above dataset into three classes using the single-link hierarchical agglomerative clustering algorithm, and draw the resulting *dendrogram*. [1 point] Is the output the same as the output of the K-means flat hierarchical clustering algorithm above? [1 point]

Yes, it is.

You maintain a text retrieval system. Let E_1 denote the complete set of documents in the index of your system and let E_2 denote the complete set of documents in the index of a competing system. Suppose the indices of both systems are independent uniform random samples without replacement from the World Wide Web N. The size of E_1 is $|E_1| = 110$ trillion $(110 \cdot 10^{12})$ documents. You take a uniform random subsample of documents without replacement from E_1 and you submit each document to the competing system. This gives you an estimate $x = 0.2$ of the conditional probability $P(d \in E_2 \mid d \in E_1), d \in N$. You repeat the same procedure with E_2, obtaining an estimate $y = 0.4$ of the conditional probability $P(d \in E_1 \mid d \in E_2), d \in N$. Assume the estimates x, y are the true probabilities. What is the size $|E_2|$ of the competing system's index? [3 points]

The grey parrot, native to equatorial Africa, is categorized as an endangered species by the International Union for Conservation of Nature (IUCN). Suppose you take a uniform random sample M without replacement of size $|M| = 8\,000$ from the grey parrot population N and mark the sampled animals. After returning the marked animals back into the population, you take a second independent uniform random sample T without replacement of the same size $|T| = 8\,000$ from the population. The number of marked animals $R = M \cap T$ in the second sample is $|R| = 10$. What is the most likely size $|N|$ of the grey parrot population? [2 points]

$$\forall d \in N: \quad P(d \in E_2 \mid d \in E_1) = x = 0{,}2$$
$$P(d \in E_1 \mid d \in E_2) = y = 0{,}4$$
$$P(d \in E_1) = |E_1| / |N|$$
$$P(d \in E_2) = |E_2| / |N|$$

$$x \cdot \frac{|E_1|}{|N|} = y \cdot \frac{|E_2|}{|N|} \rightsquigarrow |E_2| = \frac{x}{y} \cdot |E_1| = \frac{0{,}2}{0{,}4} \cdot 110 \cdot 10^{12} = 55 \cdot 10^{12}$$

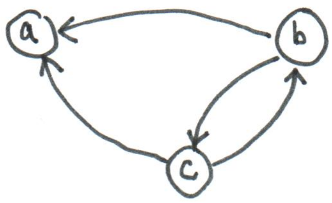
The Mark and Recapture Technique:

$$|N| = \frac{|M| \cdot |T|}{|R|} = \frac{8000 \cdot 8\,000}{10} = 64 \cdot 10^5$$

Given a directed graph G that represents three Web pages $V(G) = \{a, b, c\}$, and the links $E(G) = \{(b,a), (c,a), (c,b), (b,c)\}$ between these three pages, draw G [1 point] and produce *the adjacency matrix* (also known as *the link matrix*) $\mathbf{A}$ [1 point], and *the Markov transition matrix* $\mathbf{P}$ [2 points].

Describe the intuition behind *the PageRank algorithm* [1 point]. Compute the *PageRank* of the pages a, b, and c using a single iteration of the PageRank algorithm [2 points].

Describe what we mean, when we call a page a *hub*, or an *authority* [1 point]. Compute the *hub*, and *authority scores* of the pages a, b, and c [2 points].



Graph G:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \circ \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/2 & 1/2 & 1/2 \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \cdot (1-\alpha) + \frac{\alpha}{3}$$

where $\circ$ is the Hadamard product.

The PageRank algorithm computes the probability that a hypothetical random surfer will end up at a given web page.

$\vec{x}_0 = (1 \ 0 \ 0)$ $\quad\quad$ $\vec{x}_1 = \vec{x}_0 \cdot P = [1 \ 0 \ 0] \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ & \dots & \end{bmatrix} = [\frac{1}{3} \ \frac{1}{3} \ \frac{1}{3}]$

A hub is a web page pointing to many authorities.

An authority is a web page that many hubs point to.

$\vec{h}_0 = [1 \ 1 \ 1]^T$ $\quad\quad$ $A \cdot A^T = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}$

$\vec{a}_0 = [1 \ 1 \ 1]^T$ $\quad\quad$ $A^T \cdot A = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$

$$\vec{h}_1 = A \cdot A^T \cdot \vec{h}_0 = [0 \ 3 \ 3]^T \approx [0 \ 1 \ 1]^T$$

$$\vec{a}_1 = A^T \cdot A \cdot \vec{a}_0 = [4 \ 2 \ 2]^T \approx [1 \ \tfrac{1}{2} \ \tfrac{1}{2}]^T$$

Compute an unbiased estimate of a text retrieval system's *precision, recall*, and *the* F_1 *measure* on the first five results [2 points], and the *precision at 40% recall* [2 points] given the following lists of results for queries q_1, and q_2, where `R` is a relevant result, and `N` is a non-relevant result:

- Results for q_1 : `RNNRRNR` (10 relevant results for q_1 exist in the collection.)
- Results for q_2 : `NRNRRRRN` (5 relevant results for q_2 exist in the collection.)

The first five results for query q_1 : RNNRR

q_2 : NRNRR

$P_1 = \frac{3}{5}$ $\quad R_1 = \frac{3}{10}$ $\quad F_{1q_1} = \frac{2 \cdot 3/5 \cdot 3/10}{3/5 + 3/10} = \frac{9/25}{9/10} = \frac{10}{25} = \frac{2}{5}$

$P_2 = \frac{3}{5}$ $\quad R_2 = \frac{3}{5}$ $\quad F_{1q_2} = \frac{2 \cdot 3/5 \cdot 3/5}{3/5 + 3/5} = \frac{3}{5}$

$P@5 = \frac{3}{5}$ $\quad R@5 = \frac{9}{20}$ $\quad F_1@5 = \frac{1}{2}$

this is one possible solution, you could use macro- or micro-averaging in your computations

---

The results with 40% recall for query q_1 : RNNRRNR

q_2 : NRNR

$R_{q_1}@7 = \frac{4}{10}$ $\quad P_{q_1}@7 = \frac{4}{7}$

$R_{q_2}@4 = \frac{2}{5}$ $\quad P_{q_2}@4 = \frac{2}{4} = \frac{1}{2}$

Write your solution only on this side of the sheet!