Data Mining

Homework 5

Due: 27/1/2016, 23:59 (if late, it **has** to be submitted before the appello you're taking).

Instructions

You must hand in the homeworks electronically and before the due date and time.

Handing in: You must hand in the homeworks by the due date and time by an email to the instructor that will contain as attachment (not links!) a .zip or .tar.gz file with all your answers and subject

`[Data Mining class] Homework 5`

After you submit, you will receive an acknowledgement email that your homework has been received and at what date and time. If you have not received an acknowledgement email within 2 days after you submit then contact the instructor.

The solutions for the theoretical exercises must contain your answers either typed up or hand written clearly and scanned.

The solutions for the programming assignments must contain the source code, instructions to run it, and the output generated (to the screen or to files).

For information about collaboration, and about being late check the web page.

Most of the questions are not very hard but require time and thought. **You are advised to start as early as possible, to work in groups, and to ask the instructor in case of questions.**

Problem 1. Let D the domain (or the universe) of n distinct objects, and let P be the set of distinct pairs of objects in D. Also, let σ_1, σ_2 be two rankings (permutations) of the elements in D. The *Kendall's tau distance* between two permutations is defined as follows: For each distinct pair $\{i,j\} \in P$ if i and j are in the same order in σ_1 and σ_2, then $K_{ij}(\sigma_1,\sigma_2) = 0$; if i and j are in the opposite order (such as i being ahead of j in σ_1 and j being ahead of i in σ_2), then $K_{ij}(\sigma_1,\sigma_2) = 1$. The Kendall's tau distance between σ_1 and σ_2 is given by $K(\sigma_1,\sigma_2) = \sum_{\{i,j\}\in P}(\sigma_1,\sigma_2)$.

Very often, instead of observing the whole ranking of the n objects we see only the sorted lists of the first k elements of the ranking. We call such list a top-k list. Let τ_1 and τ_2 be the top-k lists of two rankings of the elements in D. Then, we define the p-Kendall tau distance between τ_1 and τ_2 as follows. For a pair of objects $i,j \in D$ we consider the following cases:

1. If i and j both appear in τ_1 and τ_2 and are in the same order (such as i being ahead of j in both top-k lists), then $K^p_{ij}(\tau_1,\tau_2) = 0$.

2. If i and j both appear in τ_1 and τ_2, but in opposite order (such as i being ahead of j in τ_1 and j ahead of i in τ_2) then, $K^p_{ij}(\tau_1,\tau_2) = 1$.

3. If i and j both appear in one top-k list (say τ_1) and exactly one of i or j, say i, appears in the other top-k list (say τ_2), then if i is ahead of j in τ_1, then $K^p_{ij}(\tau_1,\tau_2) = 0$. Otherwise, $K^p_{ij}(\tau_1,\tau_2) = 1$. Intuitively, we know that i is ahead of j as far as τ_2 is concerned, since i appears in τ_2, but j does not.

4. If i, but not j, appears in one of the top-k lists (say τ_1) and j but not i appears in the other top-k list (say τ_2), then $K^p_{ij}(\tau_1,\tau_2) = 1$. Intuitively, we know that i is ahead of j as far as τ_1 is concerned and j is ahead of i as far as τ_2 is concerned.

5. If i and j both appear in one top-k list (say τ_1), but neither i nor j appears in the other top-k list (say τ_2). We call such pairs special pairs and we define $K_{ij}^p(\tau_1,\tau_2) = p$ with $0 \leq p \leq 1$.

We define the *p-Kendall tau distance* between two top-k lists to be: $K^p(\tau_1,\tau_2) = \sum_{\{i,j\} \in P_{\tau_1 \cup \tau_2}} K_{ij}^p(\tau_1,\tau_2)$, where $P_{\tau_1 \cup \tau_2}$ is the set of distinct pairs $\{i,j\} \in D_{\tau_1} \cup D_{\tau_2}$ (note that D_{τ_1} (D_{τ_2}) is the subset of elements from D that appear in τ_1 (resp. τ_2)). You are asked to prove the following:

1. Prove that the Kendall's tau distance between two permutations σ_1 and σ_2, denoted by $K(\sigma_1,\sigma_2)$ satisfies the triangle inequality.

2. Find the values of p for which the p-Kendall tau distance, K^p, satisfies the triangle inequality.

Problem 2. Use the 20 newsgroup dataset as we used it in class, e.g., look into the clustering notebook and dimensionality-reduction notebooks.

Now pick g groups (with $g = 2, 4, 8, 16, 20$ and extract the top 2000 words with highest average tf-idf score. Consider this words constituting a dictionary of n terms (words) $\mathcal{T} = \{t_1, \ldots, t_n\}$. Each term t_i is associated with its importance $w(t_i)$, defined as its average tf-idf score in the collection of documents. Now in the collection of documents each document d_i uses a subset of terms in the dictionary (i.e., $d_i \subseteq \mathcal{T}$). Consider the problem of finding a collection of k documents $\mathcal{C} \subseteq \mathcal{D}$ that cover the most important terms in the dictionary. That is, select k documents $\mathcal{C}$ such that the weighted sum of the terms covered by at least one document in $\mathcal{C}$ is maximized. In other words, find $\mathcal{C}$ such that $F(\mathcal{C}) = \sum_{t_i \in (\cup_{d \in \mathcal{C}})} w(t_i)$ is maximized.

1. Prove that function $F()$ is monotone and submodular

2. Prove that there exists a constant factor, polynomial-time approximation algorithm for this problem.

3. Implement this algorithm and show your results as follows:

   - For fixed g (and for all g's) show the value of the objective function F as a function of k.
   - For fixed g (and for all g's) and $k = 4$ show the documents that you get in your results.

Problem 3. In this problem you are requested to implement the streaming algorithms that we did in class. We will implement them on data generated by twitter. Your algorithms should be able to obtain the data and compute their estimates online.

1. Implement the Flajolet–Martin estimate for counting distinct elements (F_0). Obtain ℓ independent estimates and combine them using the median of the average technique.

2. Implement the Alon–Matias–Szegedy algorithm for estimating the second moment (F_2). Obtain ℓ independent estimates and combine the by taking the average.

3. Create programs for calculating F_0 and F_2 without the streaming model (shell commands can be useful).

First test your algorithms by trying on the dataset at:
`ftp://ita.ee.lbl.gov/traces/NASA_access_log_Jul95.gz`
by considering the frequencies of the various IPs. Experiment and report results for different values of ℓ and different group sizes (for the Flajolet–Martin schema).

Then apply your algorithms on twitter data obtained by the twitter streaming API, as they are generated. Save also the data on disk so that you can verify your algorithms.

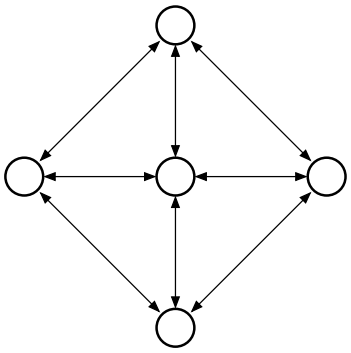
For each case, for the different values of ℓ and group sizes (for F_0) you should report in two tables, one for F_0 and one for F_2:

- the number of records
- the values of F_0 (or F_2) returned by your streaming algorithm
- the true values F_0 (or F_2)
- the absolute and relative errors
- the value of ℓ
- the group size (for the Flajolet–Martin schema)

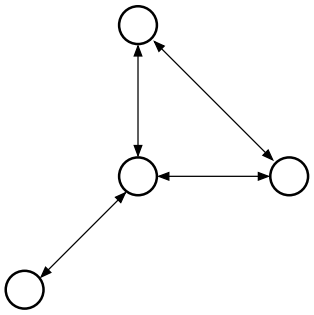
Problem 4. Here we will compute the PageRank in a special case and we will prove a rule.

1. Compute the PageRank scores of the nodes of the following two graphs, for teleporting probability α equal to zero (i.e., $\beta = 1$), using the equations of the stationary distribution. Take advantage of the symmetries to reduce the number of unknown variables: are there nodes that we know a priori that they have the same PageRank score?

(A)



(B)



2. Notice that here we have a special case: All the edges are bidirectional and we have $\alpha = 0$. After observing the scores of the nodes that you computed in these examples, make a conjecture about the PageRank score of a node in this special case, and prove it.