CSE 5243: Homework #5

Deadline: **11:59PM on 12/04/2022**.
This is the last HW. No late submissions will be accepted.

Instructions.

We currently use a 100-point scale for this homework, but it will take 10% of your final grade.

What you should turn in:

(1) For Problem 1-3, please prepare your answers in a single pdf/word file, named as something like HW5-p1-p3.pdf. Please make sure you understand how to solve them by hand (or with calculators, if necessary).

(2) For Problem 4, please put your code and all documentations/reports in a folder named as something like HW5-p4.

(3) Put all your files (Problem 1 to 4) in a Zip file named as something like HW5.zip, and submit it to Carmen. See Problem 4 for more info.

Questions?

Please create a post on Carmen discussion areas or MS teams to get timely help from other students, the TA, and me. Everyone can benefit from first checking what have been asked previously. Please try to avoid directly sending me emails.

Problem 1 (15 points).

Consider the following matrix, where there are 4 documents (S_1-S_4) and 5 rows (or, shingles) with index from 0 to 4. Assume we have chosen two hash functions to respectively simulate two permutations over the rows: (1) h1(x) = (2x+4) mod 5; (2) h2(x) = (3x+1) mod 5, where x is the current index of a row and h1(x) or h2(x) is the new index under the permutation.

Compute the signature values for each of the 4 documents under the two hash functions.

Row	S_1	S_2	S_3	S_4
0	1	0	0	1
1	0	0	1	0
2	0	1	0	1
3	1	0	1	1
4	0	0	1	0

Table 1: Matrix for Problem 1.

Problem 2 (17 points).

Consider the database shown in Table 2. Let minsup= 4. Find all frequent sequences.

Id	Sequence
s1	AATACAAGAAC
s2	GTATGGTGAT
s3	AACATGGCCAA
s4	AAGCGTGGTCAA

Table 2: Sequence database for Problem 2.

Problem 3 (18 points).

In Table 3, each sequence comprises itemset events that happen at the same time. For example, sequence s_1 can be considered to be a sequence of itemsets $(AB)_{10}(B)_{20}(AB)_{30}(AC)_{40}$, where symbols within brackets are considered to co-occur at the same time, which is given in the subscripts. **Describe an algorithm that can mine all the frequent subsequences over itemset events.** The itemsets can be of any length as long as they are frequent. **Find all frequent itemset sequences with minsup= 3.**

Problem 4 (50 points). Programming Task on Locality-Sensitive Hashing.

Your objective in this problem is to evaluate the efficacy and efficiency of Locality-Sensitive Hashing for document similarity. You will use the Sentiment Labelled Sentence Data Set (e.g., what you used in the previous two programming assignments). For each sentence, you can begin with a binary feature vector, e.g., whether a word appears in the sentence or not. As in HW1, feel free to apply pre-processing such as stemming and filter less frequent words.

Id	Time	Items
s_1	10	A,B
	20	B
	30	A,B
	40	A,C
s_2	20	A,C
	30	A,B,C
	50	B
s_3	10	A
	30	B
	40	A
	50	C
	60	B
s_4	30	A,B
	40	A
	50	B
	60	C

Table 3: Sequences for Problem 3.

Feel free to use different features than words, such as k-gram shingles we discussed in class. For example, you could use n-grams or 3-word shingles in your feature vector (results on these may yield better results as we discussed in the lecture).

(1) Creating a Baseline: For this baseline you use the raw feature vectors (i.e., the binary feature vector for each sentence described above) and for each pair of sentences, report the **exact Jaccard similarity**. We will call this the true similarity baseline.

(2) Creating a k-minhash signature: You can use any publicly available tool or create your own minhash signature for each sentence from its corresponding feature vector following the procedure described in the lecture. You will then use the k-minhash signatures (setting k at 16 and 128 respectively) and report the estimated Jaccard similarity between every pair of sentences. **Please review the lecture slides for how to estimate Jaccard similarity under the k-minhash signature.** We will call this as the k-minhash estimate of similarity.

Your report should compare (1) and (2) along the axes of efficiency (time to finish computing the similarities) and efficacy for different values of k. For efficiency, please give the absolute running time instead of some descriptive language (e.g., "method A is faster than method B" is non-informative). For efficacy or quality, you may choose to report mean-squared error (between the estimate and the true similarity) or relative mean error (normalized by the true

similarity value). You are also expected to show a table or plot these numbers in your report to facilitate comparisons across different values of k.

What You Should Turn in:

- (1) All source code. Please make it easy to run your code. (2) A detailed README file that contains all the information about the directory, e.g., how to run your program, how to interpret the output, etc. (3) Please check previous programming tasks for other comments related to letting TA understand what you did.
- (To Carmen) A short report (no more than 2 pages in 12 pt. font), describing the underlying assumptions, the approach/procedure you took, the rationale for doing it where applicable, as well as your results (name this file report3.doc or some such thing). If you use software from somewhere else, please specify and provide references in your report. Your report should describe any further data transformations you may have had to work with these software packages.

Detailing what you did is very important even if it did not work. Describe any difficulties you may have encountered, any assumptions you are making, and any future work that can be performed to get better performance.

Acknowledgement. The dataset was originally used in the following paper:
[1] "From Group to Individual Labels using Deep Features", Kotzias et al., SIGKDD 2015.