CS246: Mining Massive Data Sets **Winter 2022**

Exam

- Your Name: ______________________________
- Your SUNetID (e.g., suxuan): ______________________________
- Your SUID (e.g. 01234567): ______________________________

I acknowledge and accept the Stanford Honor Code.

Signature: ______________________________

1. There are **15** questions in this exam; the maximum score that you can obtain is **140** points. These questions require thought, but do not require long answers. Please be as concise as possible.
2. This exam is open-book and open-notes. You may use notes (digitally created notes are allowed) and/or lecture slides and/or any reference material. However, **answers should be written in your own words**.
3. Acceptable uses of computer:
   - You may access the Internet, but you may not communicate with any other person.
   - You may use your computer to write code or do any scientific computation, though writing code is not required to solve any of the problems in this exam.
   - You can use your computer as a calculator or an e-reader.
4. Collaboration with other students is not allowed in any form. Please do not discuss the exam with anyone until after Tuesday, Mar 15.
5. If you have any clarifying questions, make a **private post on Ed**. It is very important that your post is private; if it is public, we may deduct points from your exam grade.
6. Please submit your answers here on Gradescope. You have two options to submit your answers: (1) to upload **one file per question**, in a file upload field in the last sub-question; or (2) to write your answers directly in the text fields in the sub-questions.
7. Numerical answers may be left as fractions, as decimals rounded to **2 decimal places**, or as radicals (e.g., $\sqrt{2}$).

1 Frequent Itemsets Mining I (8 points)

Given that we have items A, B, C, D, E and the following 6 baskets:

Basket1	A, B, C
Basket2	A, B
Basket3	A, B, D
Basket4	B, C, D, E
Basket5	A, B, D, E
Basket6	A, B, C, D

(a) (4 points) Use Apriori algorithm to compute all the frequent itemsets (**triples**) with support ≥ 3. Please show **all of your iterations (indicating candidate/frequent itemsets)** for full score.
Write down your answers in the form of – frequent single items: $A, B, \cdots$, candidate pairs: $AB, \cdots$, frequent pairs: $AB, \cdots$ (this example doesn't indicate the results for this question).

(b) (4 points) Now we also consider a constraint when finding the frequent itemsets. Given an itemset, we want the maximum price of all the items in the itemset minus the minimum price of all the items in the set smaller than a threshold, i.e. given $I \subseteq A, B, C, D, E$, and that I is frequent, `max(item.price for item in I) - min(item.price for item in I) < T` (`T` is the threshold). Assume the price of all items and `T` are positive. How can we efficiently find frequent items under such a constraint? Design your own algorithm and briefly explain why it works.

2 Clustering (12 points)

Given the following 6 points: $x_1 = [-1, -2]$, $x_2 = [-2, 0]$, $x_3 = [4.5, 1]$, $x_4 = [3, 1]$, $x_5 = [0, -2]$, $x_6 = [2, -0.5]$. Please complete the following tasks.

(a) (3 points) If we select x_1, x_6 as the initial centroid, please write down the simulation of the K-means clustering process with L1 distance until convergence (report the new centroids and cost for each iterations in the following table, including the assignments per step, suppose x_1 and x_3 is assigned to the first cluster and all other points to the second cluster, the assignment is represented as [0, 1, 0, 1, 1, 1]).

Iteration	Centroid1	Centroid2	Cluster Assignment	Cost
0	[-1,-2]	[2, -0.5]		
1				
2				
3				

Note: you might not need all of the rows

(b) (3 points) Run hierarchical clustering with L1 distance until there are 2 clusters (please write down the clustering steps for each iterations, along with the centroids).

(c) (2 points) If we can arbitrarily set K (the number of clusters, $K <= 6$) and randomly select K points from the set as initial centroids for each of the clusters, how to make sure that we achieve the global optima of the cost function? What is the minimum cost we can achieve? How many iterations do we need? What is K?

(d) If the goal is to cluster the data points in Figure 1 to 2 clusters, which algorithm(s) works well? Select all that applies for each of them, you don't need to provide justifications.

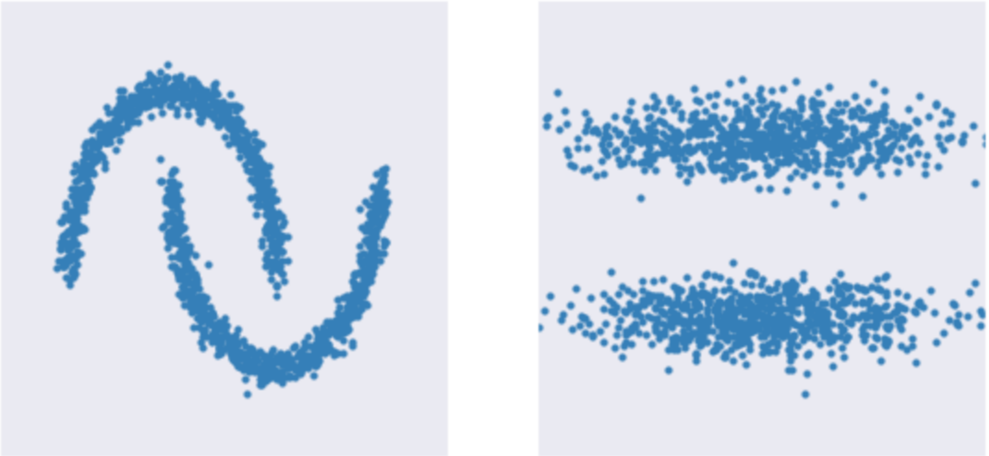


Figure 1: two example data points we want to cluster

1 (2 points) Left: K-means/ CURE/ Hierarchical clustering

2 (2 points) Right: K-means/ CURE/ Hierarchical clustering

3 Dimensionality Reduction (8 points)

We are given a matrix $A \in \mathbb{R}^{m \times n}$ and its SVD decomposition $A = U\Omega V^T$, where $U = [u_1, u_2, \cdots, u_d]$, and ($u_k \in \mathbb{R}^m$); $\Omega \in \mathbb{R}^{d \times d}$, $V = [v_1, v_2, \cdots, v_d]$, and ($v_k \in \mathbb{R}^n$). Assume Ω is sorted in descending order. Using the given information, please answer the following questions:

(a) (2 points) What is the best possible rank 1 approximation of A?

(b) (3 points) What are the eigenvalues and eigenvectors of of AA^T? (Show your steps to receive full points)

(c) (3 points) Suppose that the original matrix A is a student to CS class rating, and the latent factor is the topic of the class. The columns of matrix A are distinct courses, and the rows of matrix A correspond with the student id. For example, $A[0]$ corresponds to ratings from student id $= 0$.

i (1 point) How can you interpret $\Omega[0, 0]$?

ii (2 points) Assume Alice rated only CS246 and CS229 with a rating row vector v_a, and Bob rated only CS221 and CS224N with a rating row vector v_b; besides, the two students' ratings are not stored in A. Does it mean they are impossible to share any common interest in course topics? If it's impossible, please explain why, if possible, please explain how you can determine the similarity in their course topic interests.

4 Recommender Systems (13 points)

The table below shows ratings for three Pinterest topics by three different users.

	Kangaroos	Dragons	Pizza
User A		10	5
User B	9	16	8
User C	13		6

a) (4 points) Calculate the Pearson coefficient among all the users.

	User A	User B	User C
User A			
User B			
User C			

b) (2 points) Using Pearson correlation as similarity, fill in the missing values with User-User collaborative filtering.

	Kangaroos	Dragons	Pizza
User A		10	5
User B	9	16	8
User C	13		6

c) (4 points) Instead of user-user collaborative filtering, we would like to use latent factors to fill the missing values in the table. Given partial complete latent factor matrices Q and P^T

$$Q = \begin{bmatrix} 2 & \_\_ \\ \_\_ & 2 \\ 1 & 4 \end{bmatrix}$$

$$P^T = \begin{bmatrix} 1 & 4 & 2 \\ 3 & \_\_ & 1 \end{bmatrix}$$

Solve the missing values in Q and P such that the objective $\min_{P,Q} \sum_{(i,x)\in R}(r_{xi} - q_x p_i)^2$ is minimized to zero (r_{xi} means rating on topic i by user x). Then, solve for the missing values in the table.

Hint: $\hat{r}_{xi} = q_x \cdot p_i$, and we want to fill the missing values such that $r_{xi} - \hat{r}_{xi} = 0$ for all observed ratings.

d) (3 points) After fitting the P and Q in (c), we found that that user A is a critical reviewer such that his/her mean rating is 1 star lower than the mean, and across pinterest, Kangaroos gets a mean rating of 1 higher than average topics. Suppose the predicted rating of User A to Kangaroos is $\hat{r}_{xi} = q_x \cdot p_i = a$, where q_x and p_i are resulted from minimizing the objective function in part (c), and the overall mean rating is μ. If we take bias into account while still ignore regularization, does the new predicted rating equals to $\mu + b_x + b_i + q_x \cdot p_i = \mu + 1 + 1 + a$? Please explain your reason in two sentences.

5 Graph Neural Networks (7 points)

Suppose we have an undirected graph $G(V, E)$, consider a modified Bellman-Ford algorithm that finds the shortest distance between any node to a source node s, and we want to use GNN to learn to execute the modified Bellman-Ford algorithm (the embeddings are 1-dimensional).
At the start of the algorithm, we randomly select a source node s, and initialize the distance of any node (except for the source node) to infinity. A node becomes reachable from s if any of its neighbors are reachable from s, and we may update the shortest distance between a node v and s as the minimal way to reach any of its neighbors plus the edge length connecting the node to the neighbor.

For example, if node v_1 has two neighbors v_2 and v_3 connecting by edges of length 1, and v_1's current distance before update is $+\infty$, given $distance(v_2) = 2$ and $distance(v_3) = 4$, then we can update $distance(v_1) = 2 + 1 = 3$.

Specifically, the Bellman-Ford initialization for node embeddings is (at layer $k = 1$, for node v):

$$h_v^1 = \begin{cases} 1, & v = s \\ +\infty, & v \neq s \end{cases}$$

Similar to the example, the edges have positive weights which are the edge lengths (the edge length between node u and node v is e_{uv}). For node v, layer $k+1$, describe a message function $M(h_v^k)$, an aggregation function $h_{N(v)}^{k+1}$, and an update rule h_v^{k+1} for the GNN such that it learns the task perfectly

$$M(h_v^k) =$$

$$h_{N(v)}^{k+1} =$$

$$h_v^{k+1} =$$

6 Decision Trees (10 points)

In the decision tree learning algorithm, suppose a feature F has k different values: $v_1, v_2, \cdots, v_k$.

Let p be the total number of positive examples, and n be the total number of negative examples. Let p_i be the number of positive examples with feature value v_i, and n_i be the number of negative examples with feature value v_i.

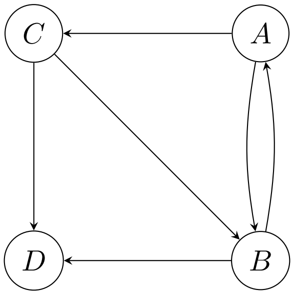
In addition, suppose

$$\frac{p_1}{p_1 + n_1} = \frac{p_2}{p_2 + n_2} = \cdots = \frac{p_k}{p_k + n_k}$$

What is the information gain if the examples are split using feature F? Show and clearly justify your derivation steps.

7 PageRank (10 points)

Suppose there are four web pages (A, B, C, D) in the network, and they form the graph as follows:



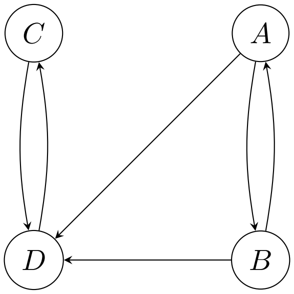
(a) (3 points) Without calculation, what is the resulting PageRank vector? Please provide justification for full scores.

(b) (4 points) Please write down column-stochastic adjacency matrix M corresponding to this graph. Given the teleport parameter β **is 0**, how can you fix the problem in the original graph through modifying M? (please write down the original M for this graph, and the resulting M after fixing this problem)

(c) (3 points) Add one edge in the graph to generate an example of spider trap, again, without calculation, what is the new pagerank vector? Why?

8 Extensions of PageRank (10 points)

Given another web graph with four web pages:



To save you some computation time, with teleport parameter $\beta = 0.8$, the resulting pagerank vector of this web graph calculated with Google matrix is: $r = (0.08, 0.08, 0.40, 0.44)$.

(a) (3 points) Suppose the user is only interested in the topic related to pageB. Given the teleport parameter $\beta = 0.8$, what is the new pagerank matrix formulation $\mathbf{A}$ for power iteration $r_{i+1} = \mathbf{A}r_i$? Please present your answer as a matrix.

(b) (2 points) (T/F) Suppose some malicious people comment thousands of blog posts with the link of their spam cite, Google can statistically process the contents (text) detect and remove duplicate pages to combat the malicious attempt. Justify your answer in one sentence.

(c) (2 points) (T/F) If we have a graph $G(V, E)$, and a set of trusted pages $T \subset V$, a page $t \in T$ will always have a non-negative spam mass. Justify your answer in one sentence.

(d) (3 points) Using the scenario of part a, suppose that our trusted page selected by the oracle is pageB (teleport parameter is still 0.8), what is the spam mass of pageD (assume power iteration algorithm converge in **1** iteration)? How does it compare to spam mass of pageC? Why is it the case? (Note: please show your work for partial credits, and you can checkout this calculator for matrix multiplications https://matrix.reshish.com/multiplication.php)

9 Graph Representation Learning (6 points)

For the following two application scenarios, determine whether Node2vec is an ideal embedding method. Explain your reasoning in 1∼2 sentences.

(a) (3 points) Learn node embeddings in a historical Facebook social network and then make predictions on a newly registered user based on its embedding.

(b) (3 points) Find functionally similar proteins across species based on the node embeddings learned in different protein-to-protein interaction networks

10 Learning Embeddings (6 points)

(a) (3 points) Suppose you have a large document dataset with millions of distinct words. In addition, the document consists of the reviews of items on Amazon, which means there are very limited numbers of rare words. Which specific model that we've seen in class would you use for feature representation of the documents? Why?
Models to choose from: Word2Vec(CBOW), Word2Vec(Skip Gram), Supervised NN with one-hot input vector, Autoencoder, SVD

(b) (3 points) What is the one similarity and one difference between PCA and autoencoder? Suppose we have a 3-layer autoencoder with input dimension of 64 and hidden dimension of 64, what would happen if we want to make predictions on unseen data?

11 Computational Advertising (10 points)

During lecture, you were given $v_i(q) = x_i(1 - e^{-f_i})$, where x_i is the bid and f_i is the fraction of leftover budget for bidder i. However, in the real world, we may want to compute $v_i(q) = x_i \times \text{CTR}_i \times (1 - e^{-f_i})$, where CTR_i is the click through rate for bidder i.

a) (3 points) There are advertisers A, B, and C with the following metrics in the table below. A new query targeted to all three advertisers arrives. Who is the winner this round?

Advertisers	Bid	CTR	Budget	Spent so far
A	120	1.2%	1000	150
B	100	1.3%	1500	200
C	80	1.9%	1200	250

b) (3 points) Continuing from part (a), there is a new query targeting all three advertisers. Who is the winner this round?

c) (4 points) There are four advertisers A, B, C, D, who bid on some of the search queries W, X, Y, Z. Each advertiser has a budget of 3 search queries. A bids on W and X only; B bids on X and Y only; C bids on Y and Z only; D bids on W and Z only. We assign queries using the BALANCE Algorithm, with ties broken alphabetically. That is, A is preferred to B/C/D, B is preferred to C/D, and C is preferred to D in case of ties. In what follows, represent the allocation of queries by a list of the advertiser given each query in order, and use a dash (-) if no bidder gets a query. For example, if A gets the first query, B the second, and no one gets the third, then write AB-.
Suppose queries arrive in the order WXYZXYZWYZWX. What is the assignment of these queries to bidders according to the BALANCE Algorithm (this question is independent from the previous parts and ties broken alphabetically)?

12 Community Detection (10 points)

We consider the following two graphs that are composed of smaller sub-graphs as building blocks. While the first graph is based on a number of *circle* graphs, the second graph connects multiple *clique* graphs.

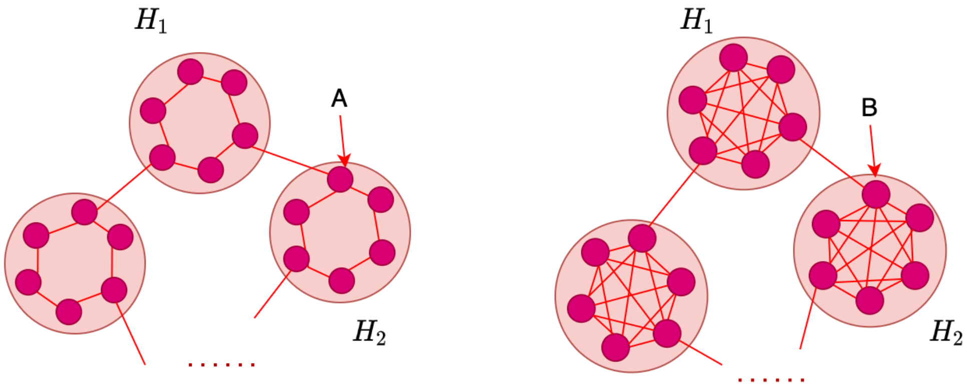


Figure 2: Graph of communities: (*left*) circle-based graph; (*right*) clique-based graph.

Let G denote the entire graph, and $H_{i=1,\cdots,K}$ denote the component circle or clique graphs. Assume there are K component graphs ($K > 2$), and each component graph contains N nodes ($N > 2$). Let $S = \{H_1, H_2, \cdots, H_K\}$ be the partitioning of G that considers each component graph as a separate community. Assume that the edge weights are all 1.

a) (6 points) For both circle-based and clique-based graph, express the total edge weights $2m$, as a function of K and N. Next, for both classes of graphs, express the modularity $Q(G, S)$, of the graph G under the partitioning S, as a function of K, N, and m.

i (3 points) Circle-based graph:

ii (3 points) Clique-based graph:

b) (6 points) We can express the modularity gain of moving nodes A and B to their adjacent community as a function of $\sum_{in}$, $\sum_{tot}$, $k_{i,in}$, k_i. Please write down the expression for the four values using K, N, m. You don't need to simplify the expression.

i (2 points) Circle-based graph $\Delta Q(A \rightarrow H_1)$:

ii (2 points) Clique-based graph $\Delta Q(B \rightarrow H_1)$:

13 Mining Data Streams (10 points)

a) (1 point) Suppose we want to calculate the number of distinct elements in the following datastream: 2, 3, 1, 1, 2, 5, 2, 1. Which of the following algorithms is best suited for this problem? (Bloom filter, Flajolet-Martin, Exponentially decaying window)
Note: You don't need to provide justification

b) (2 points) Now given a hash function $h(x) = (x + 2) \mod 64$, please use the algorithm you chose in the previous part to count the estimated number of distinct elements in the datastream.

c) (1 point) You are developing a software for an online supermarket, and your goal is to recommend new items to users, but you don't want to recommend items that they already bought. Each item has a unique long integer id, but the supermarket has so many items (2^{64} items) that you cannot store in memory. Which of the following algorithms is best suited for testing if the recommended item is unseen? (Bloom filter, Flajolet-Martin, Exponentially decaying window)
Note: You don't need to provide justification

d) (2 points) Using the algorithm you chose for the previous part, given the same hash function $h(x) = (x + 2) \mod 64$, where x is the item id, and a customer that has already bought 50 items, what is the error rate of the algorithm (labeling an item as "already bought" while the customer never bought it)?

e) (4 points) You are not satisfied with the performance of the algorithm in part d, please identify if the following methods can reduce the error rate.

i (Yes/No/Depends) Change $h(x)$ to $h(x) = (x + 2) \mod 512$ (but keeping the number of hash functions fixed and modifying them to the larger memory)

ii (Yes/No/Depends) Increasing the number of hash functions (each has the same number of possible keys/buckets as that of h)

14 Frequent Itemsets Mining II (10 points)

The idea of PCY Algorithm is: besides calculating the frequent items in the first pass, PCY also hashes pairs of items into buckets with a hash function h and calculates the frequency of pairs of items hashed into the same bucket. Then, in the second pass, we summarize the hash table we achieved in the first pass as a bitmap/bit-vector, and we can only consider candidate pairs I_{cand} that (1) consist of frequent items, and (2) are hashed into a frequent bucket (the bitmap/bit-vector of the bucket is **1**). Then, you can do the actual counting for each candidate pair.

(a) (2 points) Assume we have n items, and a hash function h that maps pairs of items to b buckets, and we use integer type (4 byte) to record the counts in the first pass. How much space (in bytes) do we need in the first pass? (Note: express your answer in terms of n and b)

(b) (2 points) In the second pass, assume we have m frequent items ($m < n$), each item ID is an integer, and we use a bit (1 byte = 8 bits) to represent whether a bucket is frequent or not, (i.e., the bitmap in the second pass), how much space (in bytes) do we need **at the start of** the second pass?

(c) Now we consider an extension to the PCY algorithm, which is called the multistage algorithm. The idea is that in the second pass, instead of doing the counting for the candidate pairs I_{cand} as in PCY, we use another hash function h' to hash I_{cand}. For pass2, we use the available main memory for another hash table, and only hash a pair $\{i, j\}$ when both i and j are frequent `AND` $\{i, j\}$ is hashed to a frequent bucket with h. Then, we will obtain a new bitmap BM2 for h'. We count pairs that are hashed to frequent buckets with **both** of h and h', the goal is to further reduce the number of candidate pairs that we do the counting.

i (3 points) Suppose we would like to utilize the benefits of multistage algorithm. Can we use the same hash function of item pairs for the first and second pass of the multistage algorithm (i.e. $h = h'$)? Explain your answer.

ii (3 points) In multistage algorithms, we can do multiple passes, in each pass we have a new hash function and further narrow down the candidate pairs. If we would like to introduce a third pass with the same process for the third pass, (note that the third pass is not the last step, we will still achieve a bitmap in the third pass, and we will have a fourth pass to do the actual counting of all the candidate pairs). Can we remove the bitmap we achieved in the first pass when constructing BM3? Explain your answer.

15 Locality-Sensitive Hashing (10 points)

Say you're trying to find similar data points in some data. Consider four data points they have: "aaa", "aaaaaa", "ababab", "aba" where we consider character level tokens.

(a) (3 points) Suppose we want to find nearby neighbors using shingling, min-hashing, and LSH. If we use 2-shingles (only consider unique shingles in a data point), 4 permutations, set $b, r = 2$ for LSH, which pairs, if any, are guaranteed (with probability 1) to always be selected as candidate pairs?

(b) (4 points) In order to align your data nicely, you decide to pad your data with the token "0" so they all have the length of the longest string. For example, "aaa" becomes "aaa000" but "aaaaaa" stays as "aaaaaa".

i In order to account for this, you decide to exclude a certain type length=2 shingles from your processing pipeline. What are the form of shingles you want to remove to make sure the candidate pairs results are unchanged even after the padding? (you can verbally describe the shingles or use regular expression)

ii Suppose you do not exclude any shingles after the padding. How likely the true candidate pairs in (a) to be identified as candidate pairs comparing to the case without padding, please think outside of the given examples? (choose from: more likely/ less likely/ the same/ unclear)?
If it is more, less, or the same, provide a proof. If it is unclear, provide counter-examples showing where it increases the chance of candidate pairs and where it decreases the number of candidate pairs being identified.

(c) (3 points) Say the data points are actually an unordered set of tokens, such as the classes of of objects detected in an image. For example, "aaabbb" is functionally equivalent to "ababab", as this indicates that 3 'a' and 3 'b' are detected (the ordering of the inputs could be arbitrary). Is there a way to modify the shingling, min-hashing, LSH pipeline such that it gives consistent results for this data and is permutation invariant? (Hint: what hyperparameters can you change?)