CS246: Mining Massive Data Sets **Winter 2025**

Exam

- Your Name: _____
- Your SUNetID: _____
- Your SUID: _____

[1 point] I acknowledge and accept the Stanford Honor Code.

Signature: _____

1. There are 12 questions in this exam, and the maximum score that you can obtain is 160 points. These questions require thought but do not require long answers. Please be as concise as possible. You can use the number of points as a rough estimate of how many minutes we think a question may take you.
2. The questions are presented in the order of which they were covered in the course. They are **not presented in order of increasing difficulty**. Therefore, we **strongly encourage** you to review the entire exam first to familiarize yourself with the layout of questions.
3. Collaboration with other students is not allowed in any form. Please do not discuss the exam with anyone until after grades are released.
4. This exam does not require the use of a calculator. Please provide your answer as a numerical value, unless otherwise indicated, in which case you may leave your answer as an expression (e.g., fractions, radicals, log terms).
5. You are only allowed a pencil and two handwritten or printed cheat sheets (front and back).
6. This exam will be curved. Do not stress if you are unable to finish every question in time. Due to the nature of this course, there is a lot of ground to cover. Prioritize solving the questions you feel the most comfortable with.
7. **Please do not write your answers in the margins of the exam - your work may not be scanned. Circle your answers or write them clearly in the designated blanks.**

Good luck!

1 Frequent Itemsets (12 points)

Spotify aims to enhance its song recommendation system by analyzing user-created playlists to identify which songs appear together frequently.

1. **(6 points)** For a collection of 6 songs in the given 6 playlists, use the **A-Priori algorithm** to find all frequent itemsets with support ≥ 3.

Playlist	Songs
1	Grape, Maple, Oxytocin, Someone
2	Grape, Maple, Oxytocin, Numb
3	Dive Back in Time, Grape, Maple
4	Someone, Numb
5	Grape, Maple, Numb
6	Dive Back in Time, Grape, Maple, Oxytocin

Please complete the table below by listing all the **candidate and frequent itemsets** for the first three iteration of the algorithm. The candidate items for the first iteration are provided as an example. The subsequent iterations correspond to pairs and triplets of items.

Note: Please refer to songs by their **first letter only**. Refer to each itemset using a {}.

Hint: This is the same algorithm you implemented in Homework 1.

Iteration	Candidate Itemsets	Frequent Itemsets
1	{G}, {M}, {O}, {S}, {N}, {D}	
2		
3		

2. **(2 points)** Suppose you apply the A-priori algorithm on a larger dataset and identify a set of frequent triplet itemsets. You then generate association rules to discover high-confidence relationships (e.g., if a user listens to X and Y, we can recommend Z).

   What is one potential downside of only selecting the **highest confidence** rules as the best recommendations? Explain your answer in 1-2 sentences.

3. **(4 points)** To improve performance, we use the **PCY (Park-Chen-Yu) algorithm**, an extension of the A-Priori algorithm that optimizes counting frequent item pairs using hashing.

   - **First Pass**: We count individual item frequencies and also hash pairs of items into buckets in a hash table.
   - **Second Pass**: We check if a pair (i, j) is frequent by ensuring both items are frequent individually and that the hash bucket the pair hashes to meets the support threshold.

   Answer the following questions in 1-2 sentences each.

   (a) **(2 points)** If a bucket meets the minimum support threshold, are all the candidate itemsets that hash to it considered frequent? Explain your answer.

   i. Yes

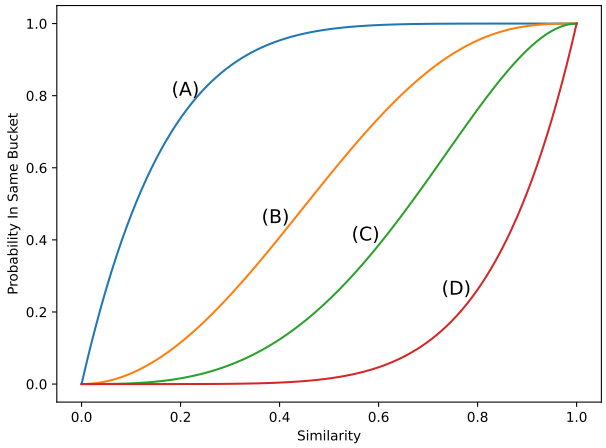
   ii. No

   (b) **(2 points)** Can the algorithm result in **false negatives**? That is, could some truly frequent pairs fail to be recognized? Why or why not?

2 Locality Sensitive Hashing (14 points)

Locality-Sensitive Hashing (LSH) is a technique for efficiently finding similar items. By using the bands-and-rows technique with MinHash, LSH balances precision and recall in similarity detection.

1. **(4 points)** The figure below illustrates different combinations of b (number of bands) and r (number of rows per band) for a total of 6 LSH functions.



(a) **(2 points)** Which of the following curves corresponds to $b = 2, r = 3$?

(A) Option A **(B)** Option B

(C) Option C **(D)** Option D

(b) **(2 points)** If we fix r and increase b, does the false positive rate increase or decrease?

(A) Increase **(B)** Decrease

2. **(4 points)** You are working on identifying near-duplicate submissions using MinHash. Suppose you are given the following words in three submissions:

$$A = \{a, b, c, d, e, f, g, h\}, \quad B = \{b, c, d, e, f, g, h, i, j\}, \quad C = \{b, f, h, m, k\}$$

You want to construct an LSH scheme with MinHash such that A and B are considered similar with probability of at least 0.85, while A and C are considered similar with a probability of at most 0.25. You decide to use 6 random MinHash functions.

(a) **(1 point)** Compute the Jaccard similarity between sets A and B.

(b) **(1 point)** Compute the Jaccard similarity between sets A and C.

(c) **(2 points)** For 6 MinHash functions, what are the optimal values for b (bands) and r (rows per band)? If it is not possible to meet the criteria in the problem statement with only 6 MinHash functions, please report that fact.

Hint: The figure above may be helpful.

Answer : $(b) =$ _____, $(r) =$ _____

3. **(6 points)** You want to ensure that near-duplicate submissions are detected properly so that plagiarism can be identified, and you need to choose a function P from one of the following:

- P_1: r-way AND followed by b-way OR
- P_2: b-way OR followed by r-way AND

For two submissions with similarity $x = 1 - \epsilon$, where $\epsilon \ll 1$, you want $P(x)$ to be as high as possible while ensuring that $r, b \geq 2$, so that P does not reduce to an AND/OR construction.

You may use, without proof, that $(1 - x)^p \approx 1 - px$ for small x.

(a) **(4 points)** Using the approximation above, calculate the approximate error $1 - P_1(x)$ and $1 - P_2(x)$ and express your answers in simplest terms using only the variables r, ϵ, b.

(b) **(2 points)** From your answers in (a), identify which of $P_1(x)$ or $P_2(x)$ you should use.

i. $P_1(x)$

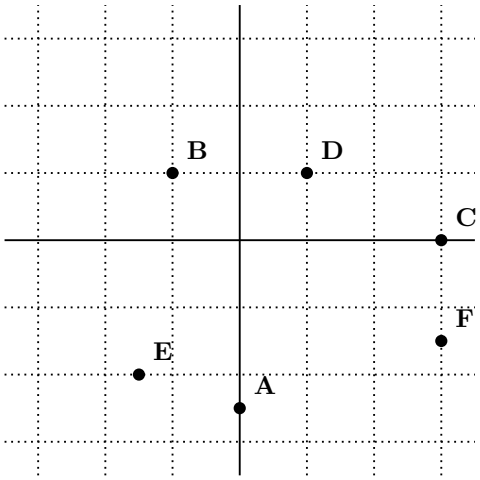
ii. $P_2(x)$

3 Clustering (12 points)

1. **(4 points)** Below is a dataset of $n = 6$ points, which we want to group into $k = 2$ clusters.

   We will use hierarchical clustering. Our criterian for the **distance between clusters** is the **maximum euclidean distance** between points from each cluster.

   Provided below is a scatterplot of the data, as well as a distance table showing the euclidean distance between pairs of points. For example, the distance between points A and B is 3.64.



(a) Graph of Points

	A	B	C	D	E	F
A	0.00	3.64	3.91	3.64	1.58	3.16
B	3.64	0.00	4.12	2.00	3.04	4.72
C	3.91	4.12	0.00	2.24	4.92	1.50
D	3.64	2.00	2.24	0.00	3.91	3.20
E	1.58	3.04	4.92	3.91	0.00	4.53
F	3.16	4.72	1.50	3.20	4.53	0.00

(b) Euclidean Distance Matrix

Apply hierarchical clustering, repeatedly merging the two closest clusters every iteration using our distance criteria, until only two clusters remain. Show the merging process.

Note: Please represent each distinct cluster as a comma-sperated bracket {}, as shown with the initialization of this algorithm in iteration 1.

Iteration Number	**Clusters**
Iteration 1	{A}, {B}, {C}, {D}, {E}, {F}
Iteration 2	
Iteration 3	
Iteration 4	
Iteration 5	

2. **(4 points)** Consider a one-dimensional space where we perform K-Means clustering on the points $1, 5, 8, 13, 19, 28$.

   Using the initial centroids 8 and 28, run K-Means to convergence and provide the **final clusters and centroids**. Format each cluster using {} for clarity.

   Clusters: _____

   Centroids: _____

3. **(2 points)** Which of the following is the most effective method to reduce K-Means' sensitivity to centroid initialization? **Select one.**

   (a) Initialize the centroids as random points in the space rather than using existing points.

   (b) Choosing initial centroids that are spread out far across the space of points.

   (c) Increasing the number of clusters.

   (d) None of the above.

4. **(2 points)** Which statement best describes the difference between BFR's Discard Set (DS) and Compressed Set (CS)? **Select one.**

   (a) DS is for points stored individually, while CS stores points in centroid-based summaries.

   (b) DS is for points sufficiently close to an existing cluster centroid; CS is for subclusters of points that are not near any existing centroid but are close to each other.

   (c) DS is "frozen" right after initialization and never updated, whereas CS keeps growing each time new data blocks are processed.

   (d) DS requires keeping a full covariance matrix for every cluster, while CS uses simpler sums of coordinates to track subclusters.

4 Dimensionality Reduction (12 points)

SVD Background.

Recall that for any real matrix $A \in \mathbb{R}^{m \times n}$, its Singular Value Decomposition (SVD) is

$$A \;=\; U \Sigma V^{\top}$$

- $U \in \mathbb{R}^{m \times m}$ is an orthogonal matrix (its columns $\mathbf{u}_1, \ldots, \mathbf{u}_m$ are the left singular vectors),
- $V \in \mathbb{R}^{n \times n}$ is an orthogonal matrix (its columns $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are the right singular vectors),
- $\Sigma \in \mathbb{R}^{m \times n}$ is diagonal (with nonnegative diagonal entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$, the singular values).

For part 1 to 3, consider the 3×2 matrix

$$A \;=\; \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 2 & 1 \end{pmatrix}. \quad \text{For convenience, we also provide} \quad A^{\top}A \;=\; \begin{pmatrix} 5 & 2 \\ 2 & 5 \end{pmatrix}.$$

1. **(3 points) Iterative Method for $\mathbf{v}_1$ and σ_1.**

   We know from lecture that the largest singular value of A can be found by computing the largest eigenvalue of $A^{\top}A$.

   Here, we use the iterative power method described in the lecture to find the principal (largest) eigenpair (eigenvector and eigenvalue) of $A^{\top}A$. With an initial guess $\mathbf{x}_0$, we use the formula:

   $$\mathbf{x}_{i+1} \;=\; \frac{(A^{\top}A)\,\mathbf{x}_i}{\|(A^{\top}A)\,\mathbf{x}_i\|},$$

   and after a few iterations, the vector $\mathbf{x}$ converges to:

   $$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

   Using the provided context, please report the value of $\mathbf{v}_1$ and σ_1.

   Value of $\mathbf{v}_1$: _____

   Value of σ_1: _____

2. **(3 points) Direct Computation of $\mathbf{u}_1$.**

   Now that we have $\mathbf{v}_1$ and σ_1, we want to find $\mathbf{u}_1$. Instead of finding $\mathbf{u}_1$ by analyzing AA^T with a similar procedure to part (1), we can directly compute $\mathbf{u}_1$ using $\mathbf{v}_1$ and σ_1 from before.

   In general, we can use the equation below:

$$\mathbf{u}_i = \frac{A\,\mathbf{v}_i}{\sigma_i} \quad (\sigma_i \neq 0)$$

   Prove that the equation above is true.

3. **(6 points) SVD vs. CUR.**

   SVD and CUR are two classic methods for low-rank matrix approximation. In this part, we explore the comparison between SVD and CUR.

   Consider a sparse matrix

$$M = \begin{pmatrix} 1 & 0 & 2 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 5 & 6 & 0 \end{pmatrix}.$$

   For a target rank $k = 2$, we obtain two different rank-2 approximations from the two methods (each rounded to two decimal points):

   - **Decomposition A:**

$$M \approx \begin{bmatrix} 0.24 & -0.95 \\ 0.24 & 0.27 \\ 0.94 & 0.17 \end{bmatrix} \begin{bmatrix} 8.24 & 0.00 \\ 0.00 & 4.33 \end{bmatrix} \begin{bmatrix} 0.03 & 0.66 & 0.74 & 0.12 \\ -0.22 & 0.39 & -0.20 & -0.87 \end{bmatrix}$$

   - **Decomposition B:**

$$M \approx \begin{bmatrix} 2.00 & 4.00 \\ 0.00 & 0.00 \\ 6.00 & 0.00 \end{bmatrix} \begin{bmatrix} 0.17 & 0.00 \\ -0.08 & 0.25 \end{bmatrix} \begin{bmatrix} 0.00 & 5.00 & 6.00 & 0.00 \\ 1.00 & 0.00 & 2.00 & 4.00 \end{bmatrix}$$

(a) **(4 points)** For the given matrix decompositions (**A** and **B**), determine which one corresponds to the result of SVD and which one corresponds to CUR.

- Decomposition **A** corresponds to: _____
- Decomposition **B** corresponds to: _____

Provide **two reasons** to justify your selection. Each explanation should be 1 sentence and be based on the properties of SVD and CUR, or relevant mathematical observations.

- **Reason 1:**
- **Reason 2:**

(b) **(2 points)** Which method provides a smaller approximation error in terms of the Frobenius norm? If this method always achieves a smaller error, why is the other method useful? Provide **one** reason and explain your answer in 1-2 sentences.

5 Music Streaming Recommender System (16 points)

You run a music streaming service with a database of songs and users. The database stores each song's duration (in seconds), genre, artist, and the number of times the song has been played by users. Users can **like** (1) or **dislike** (0) a song based on their preferences.

The table below summarizes a few songs from the database:

Song	Duration (sec)	Genre	Artist	Total Plays
S_1	240	Rock	Band X	50,000
S_2	320	Rock	Band X	120,000
S_3	180	Jazz	Artist Y	75,000
S_4	210	Pop	Artist Z	95,000
S_5	400	Classical	Orchestra W	30,000
S_6	420	Classical	Orchestra W	10,000

Table 1: Song Database

In this problem, we will use recommender systems for the task of recommending songs to users. We will work with the following models:

- User-user collaborative filtering
- Item-item collaborative filtering
- Content-based recommender system

1. **(3 points)** Consider U_1, a longtime user who primarily listens to long classical music – classical songs over 350 seconds.

   Suppose that a recommender system R suggests Song S_2 to user U_1. Which option(s) do you think R **could** be? **Circle all that apply and explain your answer in 1-2 sentences.**

   (a) User-user collaborative filtering

   (b) Item-item collaborative filtering

   (c) Content-based recommender system

2. **(3 points)** Suppose a user U_2 is interested in understanding why a particular song was recommended to them. Which of the recommendation systems outlined in the problem description would be **most effective** at providing this explanation? Justify your choice in 1-2 sentences.

3. **(7 points)** Suppose we are tasked with deciding whether to recommend an existing user U_3 a new song with **item-item** and **user-user** filtering. We have the following table with the ratings that four of our users have given to some songs (labeled S_1, S_2, etc. for convenience). We decide to recommend S_1 to U_3 if its predicted **averaged rating** over both models ≥ 0.5.

Song	U_1	U_2	U_3	U_4
Thriller (S_1)	1	0	?	1
Bohemian Rhapsody (S_2)	1	1	0	1
Stairway to Heaven (S_3)	1	1	1	0
Shape of You (S_4)	0	1	1	1
Smells Like Teen Spirit (S_5)	1	0	1	1

Table 2: Song Ratings by Different Users

We are also given two similarity matrices – one providing the pairwise similarity of user profiles and another providing the pairwise similarity between songs. They are as follows:

	U_1	U_2	U_3	U_4
U_1	1.00	0.75	0.50	0.60
U_2	0.75	1.00	0.80	0.55
U_3	0.50	0.80	1.00	0.70
U_4	0.60	0.55	0.70	1.00

	S_1	S_2	S_3	S_4	S_5
S_1	1.00	0.75	0.60	0.50	0.65
S_2	0.75	1.00	0.65	0.55	0.70
S_3	0.60	0.65	1.00	0.70	0.60
S_4	0.50	0.55	0.70	1.00	0.60
S_5	0.65	0.70	0.60	0.60	1.00

Table 3: Pairwise User and Song Similarity Matrices

(a) **(3 points)** Compute the predicted rating that user U_3 will give song S_1 as assigned by **item-item** collaborative filtering. Please consider all items provided in your calculation of the predicted rating – that is, $N = 4$ in the collaborative filtering formula.

Item-Item Rating: _____

(b) **(3 points)** Compute the predicted rating that user U_3 will give song S_1 as assigned by **user-user** collaborative filtering. Please consider all users provided in your calculation of the predicted rating – that is, $N = 3$ in the collaborative filtering formula.

User-User Rating: _____

(c) **(1 point)** Do we recommend U_3 the song S_1 – is our average predicted rating ≥ 0.5?

(A) Yes **(B)** No

4. **(3 points)** Item-item collaborative filtering tends to work better because users have multiple musical preferences. This also means that users enjoy being recommended a variety of songs.

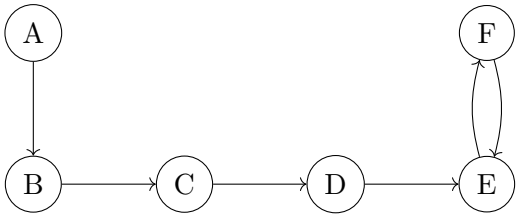
   Given the genre of each song (rock, jazz, pop) and an item-item collaborative filtering recommender system that predicts k top songs for a user, in 2-3 sentences, suggest a way to find the top 3 songs for a user such that the recommender incorporates songs from different genres.

6 PageRank and TrustRank (20 points)

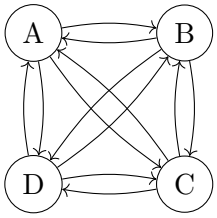
You are given three different graphs. Your task is to rank these by how quickly their PageRank values converge. Convergence occurs when each node's PageRank value stabilizes. You can assume:

- **Even initialization**: each r_i starts as $\frac{1}{n}$, where n is the total number of nodes.
- **Teleportation factor**: $1 - \beta = 0.15$.
- **Dead-end handling**: if a node has no outlinks, then with probability 1 its PageRank "teleports" (i.e., it is redistributed uniformly) from that node.

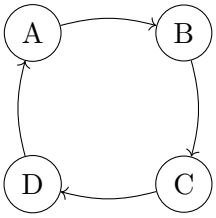
Graph A



Graph B



Graph C



Hence, the PageRank update equation you may use is:

$$r'_i = \sum_{\substack{j \text{ non-dead} \\ j \to i}} \left(\beta\, M_{ij} \;+\; (1-\beta)/n\right) r_j \;+\; \left(\tfrac{1}{n}\right) \sum_{j \text{ dead}} r_j,$$

1. **(2 points)** Which graph exhibits the fastest PageRank convergence? **Select all that apply.**

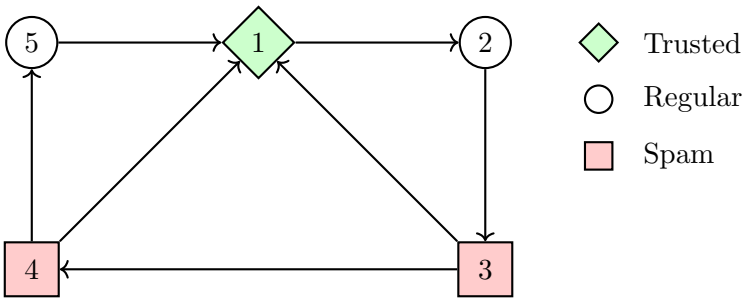
   (A) Graph A **(B)** Graph B **(C)** Graph C

2. **(2 points)** Which graph exhibits the slowest PageRank convergence? **Select all that apply.**

   (A) Graph A **(B)** Graph B **(C)** Graph C

3. **(2 points)** Rather than using teleportation, consider an alternative PageRank formulation in which we simply add a link from all dead end nodes to their parent nodes (node that points to them). In 1-2 sentences, explain whether or not this would solve the issue of dead ends without meaningfully disrupting PageRank values and provide **justification** for your answer.

4. **(14 points)** Consider the 5-node web graph shown below. Node 1 is a **trusted** page, nodes 3 and 4 are **spam** pages, and nodes 2 and 5 are webpages whose status we want to determine.



Suppose we are given the following:

- The teleport factor $(1 - \beta) = 0.2$
- All pages start with equal PageRank $(\frac{1}{5})$
- Pages with spam mass ≥ 0.5 are classified as spam

(a) **(2 points)** What is the column-stochastic adjacency matrix M for the graph above? That is, $M_{i,j}$ is the fraction of node j's outlinks that go to node i.

(b) **(4 points)** Calculate the PageRank (r_p) for all pages **after one iteration**. **Show your work** for partial credit.

r_1 : _____, r_2 : _____, r_3 : _____, r_4 : _____, r_5 : _____

(c) **(4 points)** Calculate the PageRank with teleport to trusted pages only (r_p^+) for all pages **after one iteration**. **Show your work** for possible partial credit.

r_1 : _____, r_2 : _____, r_3 : _____, r_4 : _____, r_5 : _____

(d) **(4 points)** Using part **(c)**, calculate the spam mass for nodes 2 and 5, and determine whether each node is classified as spam. Assume PageRank has converged after **one** iteration. **Show your work** for possible partial credit.

Spam Mass of Node 2: _____ **Classification of Node 2:** _____

Spam Mass of Node 5: _____ **Classification of Node 5:** _____

7 Community Detection (9 points)

Answer the following questions in 1-2 sentences each.

1. **(3 points)** In the context of graph clustering, why is conductance a better metric than just using the edge cut to evaluate the quality of a cluster?

2. **(3 points)** Recall the Louvain algorithm for community detection. Is the algorithm guaranteed to converge? If so, is it guaranteed to stop at the community assignment with maximum modularity? Explain why. If the algorithm is not guaranteed to stop, explain why not.

3. **(3 points)** Your friend did not like the PPR algorithm and Louvain algorithm. They think these algorithms are too difficult to understand. So, they proposed to use NN-Descent to do community detection. Do you think the NN-Descent algorithm can directly replace PPR or Louvain? Explain why or why not.

8 Graph Neural Networks (18 points)

1. **(2 points)** The Node2Vec algorithm introduces two key hyperparameters: the return parameter (**p**) and the in-out parameter (**q**). If $p = 1$ and $q \to \infty$, what type of structural information about the graph will the Node2Vec embeddings capture?

2. **(2 points)** Optimizing the Node2Vec objective function below is computationally expensive.

$$\max_{\mathbf{Z}} \sum_{u \in V} \sum_{v \in N_R(u)} \log P(v|\mathbf{z}_u)$$

$$P(v|\mathbf{z}_u) = \frac{\exp(\mathbf{z}_u^\top \mathbf{z}_v)}{\sum_{n \in V} \exp(\mathbf{z}_u^\top \mathbf{z}_n)}$$

   The denominator of the conditional probability is computationally expensive to compute for each node due to the outer summation over all nodes. How can we effectively mitigate this computational bottleneck? Provide a concise explanation (1-2 sentences).

3. **(2 points)** Suppose we ignore the computationally expensive term in the objective function. How might this affect the learned node embeddings? Be specific about a particular property or structure of embeddings that may change as a result.

   Note: Statements like "the embeddings will not capture similarity" will not receive credit.

4. **(4 points)** Let $h_v^{(l)}$ represent the layer l embedding of node v in some GNN with the following update rule for all l:

$$h_v^{(l+1)} = \sigma\left(W_l \sum_{u \in \mathcal{N}(v)} h_u^{(l)}\right)$$

where W_l is a trainable weight matrix and $\mathcal{N}(v)$ denotes the neighboring nodes of node v. Identify **two potential issues** when training a graph neural network with the above update function and **explain how to fix each issue** using 1-2 sentences each.

- **Reason 1:**

- **Reason 2:**

5. **(4 points)** Consider the following adjacency matrix representing a graph with nodes labeled Node 1 through Node 6.

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

(a) **(2 points)** Identify the bipartite components of the graph.
Hint: Drawing the graph represented by this adjacency matrix may help in identifying the bipartite components.

Node Numbers in Component 1: _____

Node Numbers in Component 2: _____

(b) **(2 points)** Suppose this graph is too sparse to be used in a GNN. In 1 sentence, explain what modification you can make to the graph to improve message passing.

6. **(4 points)** Stanford University stores data regarding courses in the following relational tables. The **primary key** for each table is the first, italicized column in the parentheses.

   (a) `Students` (*StudentID*, Name, Age, Major)

   (b) `Courses` (*CourseID*, Title, Department, Credits)

   (c) `Enrollments` (*EnrollmentID*, StudentID, CourseID, EnrollmentDate, Grade)

   (d) `Professors` (*ProfessorID*, Name, Department, CourseID)

   A key component of a relational deep learning framework is the **schema graph**, which illustrates how tables in a database are related. Each table is represented as a node, and whenever a table's primary key appears as a foreign key in another table, we draw a **directed edge** from the node containing the primary key to the node that uses it as a foreign key. Please draw the schema graph corresponding to these tables.

9 Decision Trees (14 points)

1. **(2 points)** When creating a decision tree to predict a binary output y based on features x, what are some methods to reduce variance and improve generalization on unseen data, without sacrificing largely on prediction quality? **Circle all that apply.**

   (a) Applying dropout with a fixed probability to each node in the tree.

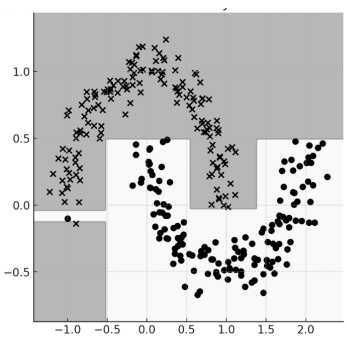
   (b) Restricting tree growth when the number of examples in the leaf is too small.

   (c) Applying ensemble learning techniques (random forests, boosting, etc).

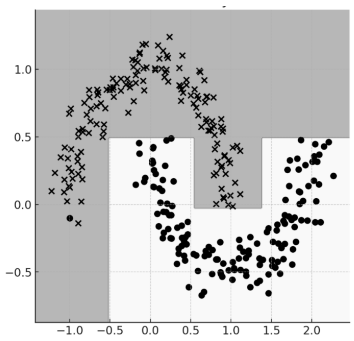
   (d) Constructing a new decision tree with only a small subset of random features.

2. **(3 points)** We create 3 different tree-based classifiers for this problem, and visualize their decision boundaries below. Suppose each point lies in two dimensional feature space (x_1, x_2). The x-axis represents values of feature x_1, and y-axis feature x_2.

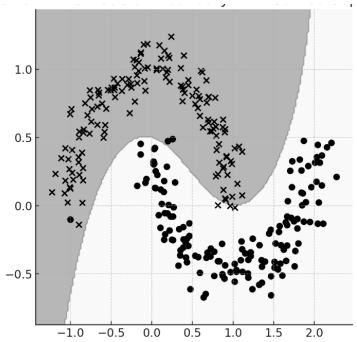
   Since this is a binary classification problem, each point belongs to one of two ground truth classes, represented by its **marker type** — either a cross or a circle. The classifier's decision boundary divides the space into two predicted classes, indicated by the dark and white regions.



(A)



(B)



(C)

Note: For each part, assume x_1 and x_2 are the only features in our model.

(a) **(1 point)** Which of the following **likely** generated **(A)**? **Circle all that apply.**

i. Decision Tree
ii. Random Forest
iii. AdaBoost
iv. None of the Above

(b) **(1 point)** Which of the following **could have** generated **(B)**? **Circle all that apply.**

i. Decision Tree
ii. Random Forest
iii. AdaBoost
iv. None of the Above

(c) **(1 point)** Which of the following **could have** generated **(C)**? **Circle all that apply.**

i. Decision Tree
ii. Random Forest
iii. AdaBoost
iv. None of the Above

3. **(6 points)** You are given a dataset containing 6 training examples. All points are at the root node, and we want to determine the **best decision boundary**, which takes the form:

$$x_i \qquad \square \qquad a$$

where:

$$i \in \{1, 2\}, \quad \square \in \{>, <\}, \quad a \in \mathbb{R}$$

The goal is to determine the correct values of i, a, and the inequality sign ($\square$) so that the split defines the **positive prediction region** ($\hat{y} = 1$). These values should be chosen to **maximize the information gain**, ensuring the best decision boundary at the root node.

In your answer, please report the following.

(a) **(3 points)** The best values of i, $\square$, and a.
There can be multiple correct answers.

(b) **(3 points)** The information gain achieved from this optimal split.
You do not need to simplify your answer.

Hint: You may find it helpful to first graph these points to visually determine the best split, rather than calculating information gain for each possible split by hand.

x_1	x_2	y
1	3	1
2	1	0
2	4	0
3	2	0
4	4	1
5	2	1

Table 4: Dataset for Decision Tree Splitting

Additional Context

The **information gain** for a split on attribute x_i at threshold a is:

$$IG = H(Y) - \left(\frac{N_L}{N} H(Y_L) + \frac{N_R}{N} H(Y_R) \right)$$

where $H(Y)$ is the entropy of the parent node, $H(Y_L)$ and $H(Y_R)$ are the entropies of the left and right child nodes after the split. N is the total number of examples, and N_L and N_R are the number of examples in the left and right child nodes, respectively.

Entropy is defined as:

$$H(Y) = -\sum_{c \in \{0,1\}} p_c \log_2 p_c$$

where p_c is the proportion of examples belonging to class c.

You do not need to simplify your Information Gain - leave it as an expression!

i : _____ $\square$: _____ a : _____

Information Gain : _____

4. **(3 points)** Which of the following are key facts about AdaBoost and Gradient Boosted Decision Trees (GBDT), like XGBoost? **Circle all that apply.**

   (a) AdaBoost assigns sample weights, while GBDT optimizes based on gradients.

   (b) AdaBoost re-weights misclassified examples more heavily at each iteration.

   (c) GBDT have built-in regularization to prevent overfitting.

   (d) GBDT mostly use weak learners that are always decision stumps.

10 Mining Data Streams (12 points)

Bloom Filter Background:

1. Create a bit array B of n bits, initially all 0s.
2. Choose k independent hash functions $H_1, H_2, \ldots, H_k$ which hash the input domain to the range $[0, n)$ uniformly at random.
3. To insert an item x, set $B[H_1(x)] = 1, B[H_2(x)] = 1, \ldots, B[H_k(x)] = 1$.
4. To check if an item y was inserted, check if $\bigwedge_{i=1}^{k} B[H_i(y)] == 1$ (i.e. all the corresponding hashed locations have 1 or not).

Bloom filters can sometimes incorrectly indicate that an item is present (false positive), but they will never incorrectly indicate that an item is absent when it was actually added (no false negatives).

Key Facts:

If we insert m distinct items in a bloom filter of size n bits which has k hash functions, then:

- $\Pr(B[i] = 1) = (1 - (1/n)^{km}) = 1 - e^{-\frac{km}{n}}, \forall i \in [1, n]$
- $\Pr(\text{false positive}) = (1 - e^{-\frac{km}{n}})^k$.

Problem Setup:

You are the network administrator for a company with two secondary servers, S_1 and S_2, each serving m distinct IP addresses during the day. These servers are located in different regions, and their IPs do not overlap. Each server maintains a Bloom filter of size n bits to track the IPs it served, using k hash functions.

At the end of the day, you need to combine the two Bloom filters into a single Bloom filter data structure of at most n bits on the primary server. This combined Bloom filter must allow you to query whether a specific IP address x was served by either server.

- Combine method: $c(b_1, b_2)$: Takes in the two bloom filters (b_1, b_2) and returns a new data structure of **size at-most n bits**.
- Query method: $q(x)$: Returns whether the IP address x was served by one of the secondary servers or not during the day. This method can access the data structure created and the hash functions $H_j^i, \forall i \in \{1, 2\}, \forall j \in [1, k]$.

Problem Statement:

For the given cases, provide the best implementation of $c(\cdot)$ and $q(\cdot)$ aimed at minimizing the false positive probability, along with the corresponding upper bound on the false positive probability. You are not required to prove that this bound is the lowest possible.

Important: Assume $k = \frac{n}{2m} \ln 2$ and $k \gg 1$ for both the cases.

Hint: Focus on a straightforward approach where you combine the information from both Bloom filters directly without overcomplicating the solution.

1. **(6 points)** Suppose S_1 and S_2 have the same set of k hash functions ($H_j^1 = H_j^2, \forall j \in [1, k]$)

2. **(6 points)** Suppose S_1 and S_2 have different hash functions, all chosen independently ($H_{1...k}^1, H_{1...k}^2$)

11 Submodular Functions (10 points)

1. **(2 points)** Suppose we want to pick $k = 3$ sets from the following collection:

$$\{\{0, 1, 4, 9\}, \{0, 6, 9\}, \{1, 2, 4, 8\}, \{3, 5, 7\}, \{9\}, \{0\}\}$$

Would a greedy algorithm that selects sets based on size (i.e. picking the largest set first) recover a set of three that covers the greatest possible number of distinct elements? Explain.

2. **(8 points)** For each of the set functions below, determine if it is submodular. If yes, explain why in one or two sentences. If no, provide a counterexample. Assume that $F : P(\mathbb{R}) \to \mathbb{R}$ where $P(\cdot)$ denotes the powerset (the set of all subsets).

Recall that a function is submodular if for all $A \subseteq B$ and some set C,

$$F(A \cup C) - F(A) \geq F(B \cup C) - F(B)$$

(a) **(2 points)** $F(A) = |A|$ where $|\cdot|$ denotes the size of the set.

(b) **(2 points)** $F(A) = \max(A)$. Let $\max(\emptyset) = 0$.

(c) **(2 points)** $F(A) = \sum_{i \in A} i^2$

(d) **(2 points)** $F(A) = (\sum_{i \in A} i)^2$

12 Bandits (10 points)

The multi-armed bandit problem is a framework for sequential decision-making under uncertainty, capturing the trade-off between exploration (gathering information about unknown rewards) and exploitation (leveraging the best-known option). Each "arm" represents a different action or choice with an initially unknown reward distribution. Through repeated interactions, the algorithm learns about the arms' rewards and refines its strategy to maximize cumulative payoff.

1. **(2 points)** [T/F] In multi-armed bandit algorithms, the value of the **regret** metric is unknown while the algorithm is running and can only be determined after the process finishes.

   (a) True (b) False

2. **(2 points)** [T/F] The Upper Confidence Bound (UCB) algorithm will not pick suboptimal arms once it finds an arm that appears optimal.

   (a) True (b) False

3. **(2 points)** [T/F] Statistical tests (e.g., t-tests) used in A/B testing can be applied to assess the statistical significance of one variant's superiority in a multi-armed bandit experiment.

   (a) True (b) False

4. **(4 points)** A gambler is playing a 3-armed bandit machine, where each arm has an unknown probability of giving a reward of 1, otherwise giving reward 0.

   The gambler follows an ϵ-**greedy** algorithm with $\epsilon = 0.3$. That is, with an 70% probability, they exploit (choose the best-known arm so far), and with a 30% probability, they explore (choose an arm randomly).

   The gambler starts with no prior knowledge of the probabilities and plays 10 rounds. Suppose they observe the following sequence of rewards:

Round	Chosen Arm	Reward
1	1	1
2	2	0
3	3	1
4	1	0
5	2	1
6	3	0
7	1	1
8	2	1
9	3	0
10	2	1

Table 5: Rounds, chosen arms, and rewards

(a) **(2 points)** Compute the estimated **mean reward** for each arm based on the data.

Mean Reward of Arm 1: _____

Mean Reward of Arm 2: _____

Mean Reward of Arm 3: _____

(b) **(2 points)** Under the ϵ-greedy algorithm with $\epsilon = 0.3$, what are the probabilities of choosing each arm for the next round?

Probability of Choosing Arm 1: _____

Probability of Choosing Arm 2: _____

Probability of Choosing Arm 3: _____