CS246 Exam Review

Fall 2022

1. There are **15** questions in this exam; the maximum score that you can obtain is **130** 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 Sunday, Mar 13. The new ddl is 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 Dimensionality Reduction [10 points]

Consider the following 3×3 matrix:

$$M = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

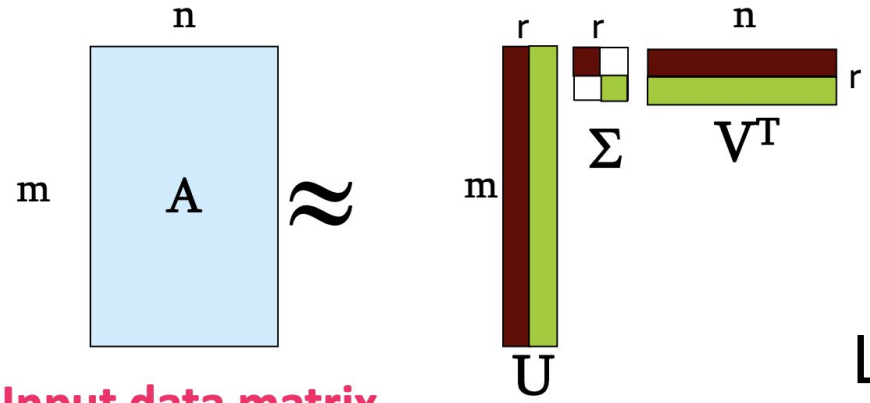
The SVD of M is given as follows:

$$M = U\Sigma V^T = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(a) **Singular Values**: What are the singular values of M? ★ **Solution** ★ 4, 2, and 1.

SVD – Definition

$$\mathbf{A} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i^T$$



Lecture 6, page 10

- **A**: **Input data matrix**
  - *m* x *n* matrix (e.g., *m* documents, *n* terms)
- **U**: **Left singular vectors**
  - *m* x *r* matrix (*m* documents, *r* concepts)
- **Σ**: **Singular values**
  - *r* x *r* diagonal matrix (strength of each 'concept') (*r* : rank of the matrix **A**)
- **V**: **Right singular vectors**
  - *n* x *r* matrix (*n* terms, *r* concepts)

(b) **Low-rank Approximation**: We wish to obtain N, which is the **best possible rank 2 approximation** of M (in terms of reconstruction error based on the Frobenius norm). Remember that N must also be a 3×3 matrix. Calculate N and its singular value decomposition.
Note: Frobenius norm of a matrix A is defined as follows: $||A_F|| = \sqrt{\sum(A_{ij}^2)}$

★ **Solution** ★ We use the SVD to minimize reconstruction error based on the Frobenius norm.

$$N = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

(c) **Reconstruction Error**: The reconstruction error between two matrices is defined as the Frobenius norm of their difference. What is the reconstruction error between M and N?

★ **Solution** ★ 1

2 MapReduce [10 points]

In Gradiance quizzes and homework you have seen matrix-vector multiplication in MapReduce. This time you will work on a slightly different task. Given two sets

$$R = \{a, b, c\}, S = \{b, e, f\},$$

we want to find the difference of $R \setminus S$, i.e., the set of elements that exists in R but not in S. Design Map, Group by Key and Reduce functions to compute the set difference, and write your answers below in terms of a, b, c, e, f, R, S.

★ Solution ★

(a) $R : (a, R), (b, R), (c, R), S : (b, S), (e, S), (f, S)$;

(b) $(a, [R]), (b, [R, S]), (c, [R]), (e, [S]), (f, [S])$;

(c) $\{a, c\}$

Or any other reasonable answer.

3 Frequent Item Set Mining [10 points]

Suppose we are given some documents of different domains or topics, and we would like to categorize them according to the words in the documents. Specifically, we treat documents as baskets, and words as items. Your goal is to find the frequent item sets, and then categorize the item sets, so that documents can be assigned to different categories.

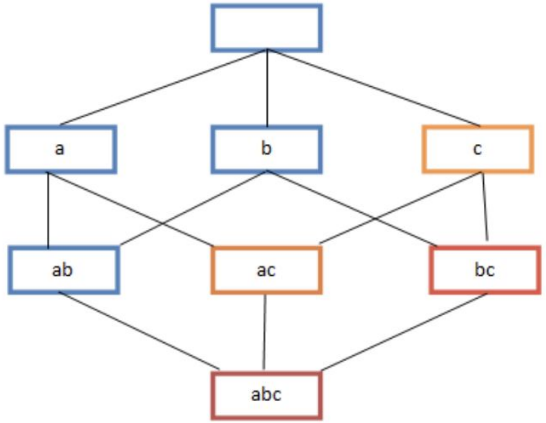
In this problem we focus on the frequent item set mining part. As a toy example, assume the whole item set is $S = \{\text{banana, apple, basket, friend, atmosphere, learning}\}$. (An example of document can be "The apple is in the basket".) And we have the following table of baskets and items:

Sentence index	words in the sentence and S
1	banana, apple, basket
2	basket, learning
3	apple, friend, learning
4	basket, friend, atmosphere
5	banana, friend, atmosphere, learning
6	basket, friend, atmosphere

Consider **support threshold** $s = 3$ in this case. Apply the A-priori algorithm to find the frequent item sets.

A-Priori Algorithm – (1)

- A **two-pass** approach called ***A-Priori*** limits the need for main memory
- **Key idea: *monotonicity***
  - If a set of items I appears at least s times, so does every **subset** J of I



- **Contrapositive for pairs:**
  If item i does not appear in s baskets, then no pair including i can appear in s baskets

For a bucket b with count c, what can you say about the pairs that hash to b if $c \geq s$? (Possible answers: They must be frequent, they cannot be frequent, or not sure.)

What can you say if $c < s$? (They must be frequent, they cannot be frequent, or not sure.)

How are these counts stored when doing the second pass? (one line answer is sufficient)

(a) $L_1 = \{\text{basket, friend, atmosphere, learning}\}$

(b) $C_2 = \{\{\text{basket, friend}\}, \{\text{basket, atmosphere}\}, \{\text{basket, learning}\}, \{\text{friend, learning}\}, \{\text{friend, atmosphere}\}, \{\text{atmosphere, learning}\}\}$

(c) $L_2 = \{\{\text{friend, atmosphere}\}\}$

(d) Not sure.
They cannot be frequent.
bitmap where one bit for each bucket (1 if count is more than s)

Observations about Buckets

- **Observation: If a bucket contains a frequent pair, then the bucket is surely frequent**
- However, even without any frequent pair, a bucket can still be frequent ☹

Lecture 2, page 41

  - So, we cannot use the hash to eliminate any member (pair) of a "frequent" bucket
- **But, for a bucket with total count less than s, none of its pairs can be frequent ☺**
  - Pairs that hash to this bucket can be eliminated as candidates (even if the pair consists of 2 frequent items)

5 Machine Learning Memes for Decision Trees [10 points]

Feature selection in regression

Like most people, you are fond of memes. You decide to build a decision tree to predict the number of likes y_i for a given post i on the Stanford meme page. Your training dataset has 1000 posts. The variance of y_i in your training dataset is 500. You have been given a few candidate features and want to figure out which is the best one to split on. Let $|D_L|$ and $|D_R|$ represent the size of the left and right child datasets after splitting. Let $Var(L)$ and $Var(R)$ represent the variance of y_i in the left and right child datasets after splitting.

(a) Which of these features would you choose for splitting at the top level, and why?

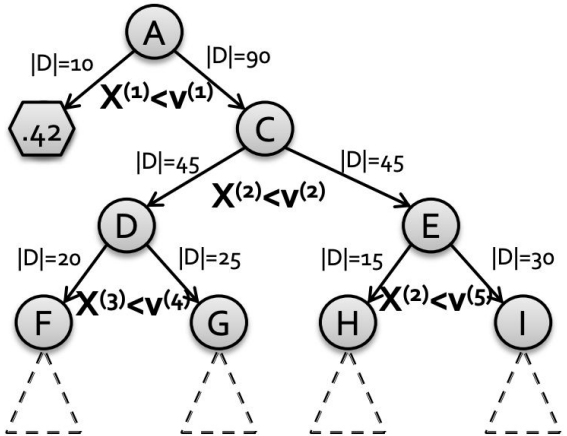
- Word count: $|D_L| = 800$, $|D_R| = 200$, $Var(L) = 600$, $Var(R) = 100$
- Content related to machine learning: $|D_L| = 300$, $|D_R| = 700$, $Var(L) = 100$, $Var(R) = 600$
- Number of prior posts by user: $|D_L| = 400$, $|D_R| = 600$, $Var(L) = 100$, $Var(R) = 700$

★ **Solution** ★ Content related to machine learning.
We want to maximize $|D| \times Var(D) - (|D_L| \times Var(D_L) + |D_R| \times Var(D_R))$.

How to construct a tree?

(1) How to split? Pick attribute & value that optimizes some criterion

- **Regression: Purity**
  - **Find split** $(X^{(i)}, v)$ that creates D, D_L, D_R: parent, left, right child datasets and **maximizes**:



$$|D| \cdot Var(D) - \left(|D_L| \cdot Var(D_L) + |D_R| \cdot Var(D_R)\right)$$

  - $Var(D) = \frac{1}{|D|}\sum_{i \in D}(y_i - \bar{y})^2$ … variance of y_i in D

Bagging: Now consider a smaller dataset D_1 with only 3 examples:

Word Count	ML Content	Prior Posts	Likes
5	Yes	4	500
10	Yes	6	400
15	No	8	100

Your friends Leland and Stanford want to use bagging to improve model performance. They use D_1 to generate synthetic datasets for bagging. Leland's dataset D_1' looks like this (table below):

Word Count	ML Content	Prior Posts	Likes
15	No	8	100
5	Yes	4	500
15	Yes	10	900

Stanford's dataset D_1'' looks like this (table below):

Word Count	ML Content	Prior Posts	Likes
15	No	8	100
5	Yes	4	500
15	No	8	100

(b) Which of the two has an incorrect implementation of bagging, and why? ★ **Solution** ★

Leland. The third example in D_1' does not occur in the original dataset D_1.

Why Information Gain?

- **Suppose I want to predict Y and I have input X**
  - **Def: Information Gain**
    - $IG(Y|X)$ = I must transmit Y. **How many bits on average would it save me if both ends of the line knew X?**

$$IG(Y|X) = H(Y) - H(Y \mid X)$$

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
Math	Yes
History	No

- **Example:**
  - H(Y) = 1
  - H(Y|X) = 0.5
  - Thus **IG(Y|X) = 1 – 0.5 = 0.5**

Lecture 13, page 27

IG tells us how much information about Y is contained in X

6 Clustering [10 points]

(a) In this question, you will perform a simple K-means calculation.

Points	x	y
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

If we set $k = 2$, the initial centroids be **P1** and **P4**, and we are using Euclidean distance, what are the final outputs of K-means?

(b) *True/False* question about K-means.
If we use **Euclidean distance**, the cost over iterations always decreases: _____

(c) Explain two different types of 2-dimensional data distributions where K-means might fail to produce accurate clusters. Provide sketches of the data in the 2D Euclidean space, and briefly explain.

(a) Cluster1: {1, 2}, (1.25, 1.5); Cluster2: {3, 4, 5, 6, 7}, (3.9, 5.1);

(b) True before converging and False after convergence. So both are accepted;

(c) Clusters with outliers; Clusters with different densities; clusters of non-convex shape, e.g. concentric clusters; Other answers and sketches that make sense also counts, but 2 cases should be presented, while in a lot of submissions only one is presented;

Hierarchical vs point-assignment

- **Point assignment good when clusters are nice, convex shapes:**

- **Hierarchical can win when shapes are weird:**
  - Note both clusters have essentially the same centroid.

Lecture 5, slide 17

Notes:

- Please make sure you understand the advantages/drawbacks of clustering algorithms we covered in class
- CURE, Hierarchical, …

7 Advertising [10 points]

During the lecture, you were given $\psi_i(q) = x_i(1 - e^{-f_i})$, where x_i is the bid and f_i is the fraction of left over budget for bidder i. In reality, sometime you would probably want to compute $\psi_i(q) = x_i CTR_i(1 - e^{-f_i})$, where CTR_i is the click through rate for bidder i.

(a) How do you interpret the new formula for ψ_i against the original form (short answer in 1 sentence)?

(b) Suppose you have the following table for 3 advitisers. From now on, use the new ψ formula as your metric to choose advertisers.

Advertisers	Bid	CTR	Budget	Spent so far
A	50	2%	1000	100
B	80	1.5%	2000	250
C	50	2.5%	1500	300

A new query targeted for all three advertiser arrived. Who is the winner?

(c) Following up from (b), the system received a new query targeted for all three advertiser. Who is the winner at this round?

(d) Give a reasonable bound for the algorithm used in this problem (answers like ≤ 1 will not be awarded). Why is this form preferable compared to the one given in the lecture?

(a) The new form computes expected return times the trade off function whereas the original form only times bid with the trade off function.

(b)

$$\psi_A = 0.59, \psi_B = 0.6997, \psi_C = 0.688.$$

Choose B.

(c)

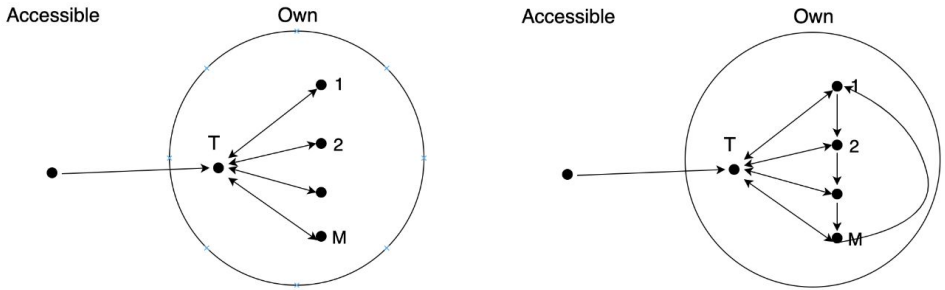
$$\psi_A = 0.59, \psi_B = 0.679, \psi_C = 0.688.$$

Choose C.

(d) Competitive ratio is less than $(1 - 1/e)$ since we can not beat the optimal competitive ratio. The optimal competitive ratio assumes that we have infinitely many queries for given set of budgets and bid prices. In reality we need to take time into account and maximize gain within a time window.

8 Link Spam [10 points]

Consider two spam farm structures shown below. Spam farm 1 is the example you studied in class, and spam farm 2 is a modified version of the given example. For both structures, node T is the target page, and the spammer hopes to maximize its page rank y. The spammer owns M farm pages and has 1 accessible page with only one out link linking to T. The accessible page has page rank a. There are N pages in the entire web. The teleportation parameter is β. The only difference between the two structures is that in spam farm 2, each farm page has one additional out link pointing to another farm page, and these out links form a directed cycle as shown in the graph.



(a) Spam Farm 1 (b) Spam Farm 2

Figure 2: Two Spam Farms

(a) For spam farm 2, write down the PageRank equations to compute the PageRank score of the farm page (f) and the PageRank score of the target page (y) (in terms of f, y, a, β, M, and N)

Link Spamming

- **Three kinds of web pages from a spammer's point of view**
  - **Inaccessible pages**
  - **Accessible pages**
    - e.g., blog comments pages
    - spammer can post links to his pages
  - **Owned pages**
    - Completely controlled by spammer
    - May span multiple domain names

Lecture 10, page 51

Note: for this topic, please also review link analysis, and methods to **combat spam on the web**

- PPR
- TrustRank

- **PageRank equation** [Brin-Page, '98]

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1 - \beta) \frac{1}{N}$$

- **The Google Matrix *A*:**

$[1/N]_{NxN}$…N by N matrix where all entries are 1/N

$$A = \beta M + (1 - \beta) \left[\frac{1}{N}\right]_{N \times N}$$

Lecture 9, page 52

(b) Which spam farm out of the two do you think will yield higher page rank for T? Please explain your reasoning.

★ Solution ★

(a)

$$f = \beta \frac{y}{M} + \frac{\beta f}{2} + \frac{1-\beta}{N} \tag{1}$$

$$y = \beta a + \beta \frac{f}{2} M + \frac{1-\beta}{N} \tag{2}$$

(b) a will have higher page rank. In b some of the page rank is shared by the spam farm due to the loop.

9 Collaborative Filtering [10 points]

Consider the table of ratings below, which shows the ratings for three different books by five different readers. In this question, you want to figure out the rating of Reader1 for Book1 using item-item and user-user collaborative filtering methods. Notice that some of the ratings are unknown.

	Book1	Book2	Book3
Reader1	?	2.0	1.0
Reader2	3.0	1.0	
Reader3	1.0		
Reader4	2.0	1.0	
Reader5	0.0		3.0

(a) Use the user-user collaborative filtering method and the **cosine similarity** measure to calculate the rating of Reader1 for Book1 based on the **two** most similar readers to Reader1. Show your steps and highlight your final rating. **You do not need to subtract row mean to normalize the table**.

(b) Use the item-item collaborative filtering method and the **cosine similarity** measure to calculate the rating of Reader1 for Book1 based on the **most** similar book to Book1. Show your steps and highlight your final rating. **You do not need to subtract row mean to normalize the table**.

★ Solution ★

(a) sim(R1,R2) = 0.283
sim(R1,R3) = 0
sim(R1,R4) = 0.4
sim(R1,R5) = 0.447
rating = $(2.0 \times 0.4 + 0.0 \times 0.447)/(0.4 + 0.447) = 0.9445$

(b) sim(B1, B2) = 0.5455
sim(B1, B3) = 0
rating = 2.0

Lecture 7, page 24

Note: (Pearson correlation) the numerical example on page 30 is inconsistent with this page, as page 30 uses the norm of r_x and r_y (minus mean) for normalization, while the definition uses the the norm of r_x', r_y' (their entries containing non-zero locations for both r_x and r_y).

Related to the example on the right
r_x' = (1-5/3, 1-5/3) = (-0.67, -0.67)
r_y' = (1-5/3, 2-5/3) = (-0.67, 0.33)

Finding "Similar" Users

r_x = [1, _, _, 1, 3]
r_y = [1, _, 2, 2, _]

- Let r_x be the vector of user x's ratings
- **Jaccard similarity measure**
  - **Problem:** Ignores the value of the rating

r_x, r_y as sets:
r_x = {1, 4, 5}
r_y = {1, 3, 4}

- **Cosine similarity measure**
  - $\text{sim}(x, y) = \cos(r_x, r_y) = \frac{r_x \cdot r_y}{||r_x|| \cdot ||r_y||}$
  - **Problem:** Treats some missing ratings as "negative"

r_x, r_y as points:
r_x = {1, 0, 0, 1, 3}
r_y = {1, 0, 2, 2, 0}

- **Pearson correlation coefficient**
  - S_{xy} = items rated by both users x and y

$$sim(x, y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \overline{r_x})(r_{ys} - \overline{r_y})}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \overline{r_x})^2} \sqrt{\sum_{s \in S_{xy}} (r_{ys} - \overline{r_y})^2}}$$

$\overline{r_x}$, $\overline{r_y}$ … avg. rating of x, y

10 Latent Factors [10 points]

In this question, rather than using collaborative filtering, you will use a basic latent factor model for making recommendations. For this model, the predicted rating $\hat{r}_{xi}$ for user x on item i is computed as follows:

$$\hat{r}_{xi} = q_i \cdot p_x$$

where q_i is row i of matrix Q, and p_x is row x of matrix P. Consider the incomplete ratings matrix R in the table below, where r_{xi} is the true rating for user x on item i, along with the partially completed latent factor matrices Q and P^T.

	User1	User2	User3	User4
Item1	1.0		4.5	
Item2		5.0		2.0
Item3	1.5			3.0
Item4		2.5	1.5	

$$Q = \begin{bmatrix} 2 & \text{---} \\ \text{---} & 2 \\ 1 & \text{---} \\ \text{---} & 1 \end{bmatrix}$$

$$P^T = \begin{bmatrix} -0.5 & 1 & \text{---} & \text{---} \\ 2.0 & \text{---} & 1.5 & 1 \end{bmatrix}$$

(a) Fill in the missing entries of Q and P^T (denoted by ---) such that $r_{xi} - \hat{r}_{xi} = 0$ for all observed ratings. Please rewrite both matrices fully in the space given below.

(b) Using the completed matrices from part (a), fill in the unobserved ratings in the ratings matrix for User 4 with predictions generated from the model.

(a)

$$Q = \begin{bmatrix} 2 & 1 \\ 0 & 2 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$P^T = \begin{bmatrix} -0.5 & 1.0 & 1.5 & 2.0 \\ 2.0 & 2.5 & 1.5 & 1.0 \end{bmatrix}$$

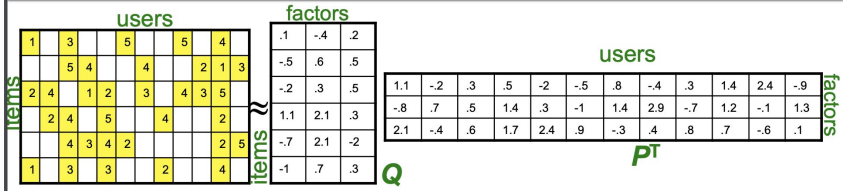
(b) Only values for User 4 must be completed.

	User1	User2	User3	User4
Item1	1.0	4.5	4.5	5.0
Item2	4.0	5.0	3.0	2.0
Item3	1.5	3.5	3.0	3.0
Item4	2.0	2.5	1.5	1.0

Objective function
What is the lower bound for it?

Lecture 8, page 28

Latent Factor Models



- SVD isn't defined when entries are missing!
- Use specialized methods to find *P, Q*
  - $\min_{P,Q} \sum_{(i,x)\in R} (r_{xi} - q_i \cdot p_x)^2$ $\quad \hat{r}_{xi} = q_i \cdot p_x$
  - **Note:**
    - We don't require cols of *P, Q* to be orthogonal/unit length
    - *P, Q* map users/movies to a latent space
    - This was the most popular model among Netflix contestants

11 Bloom Filter [10 points]

Suppose you are building a Youtube recommendation system. Each video on Youtube is represented by a unique 64-bit index. You have already developed recommendation algorithm which would give us a set R of videos a particular user might like. But you certainly don't want to recommend a video already watched by that user. So you want to test whether a video returned by your recommendation algorithm is in the watch history of that user or not. Clearly, a user might have seen so many videos that you can't afford storing all the video identifiers in memory. Therefore, you decide to use a bloom filter data structure to achieve the goal.

(a) Suppose the user has watched a total of 1000 videos. You want to construct a bloom filter with a bit array B of m bits, using a hash function $h : \{0, 1, 2, ..., 2^{64}-1\} \rightarrow \{0, 1, 2, ..., m-1\}$. The minimum value of m to achieve a false positive probability of 0.1 for your bloom filter will be _____. (Write down your answer as an integer) ★ **Solution** ★

9491. 9492 is also fine.

(b) Now, suppose you are not satisfied with the 0.1 false positive probability for your bloom filter, and want to achieve lower false positive probability. Suggest two different ways to modify the design of the bloom filter in part (a) to achieve your goal. How will your approaches affect the false negative probability? (For each approach, give 1-2 sentence(s) description and 1-2 sentence(s) explanation of why it would decrease false positive probability, and how it would affect false negative probability.)

★ **Solution** ★ 1. Increase the memory of our bloom filter will help decrease false positive rate. The student can either explain using math formula or give an intuitive explanation. The false negative rate will remain the same (zero).
2. Increase the number of hash functions will also help. Since the optimal number of hash functions k is m/1000(ln2) while we are only using one hash function right now, more hash functions gonna help us avoid collision. The false negative rate will remain the same (zero).

Bloom Filter – Analysis

- **What fraction of the bit vector B are 1s?**
  - Throwing ***k·m*** darts at ***n*** targets
  - So fraction of **1**s is $(1 - e^{-km/n})$

Lecture 15, page 33

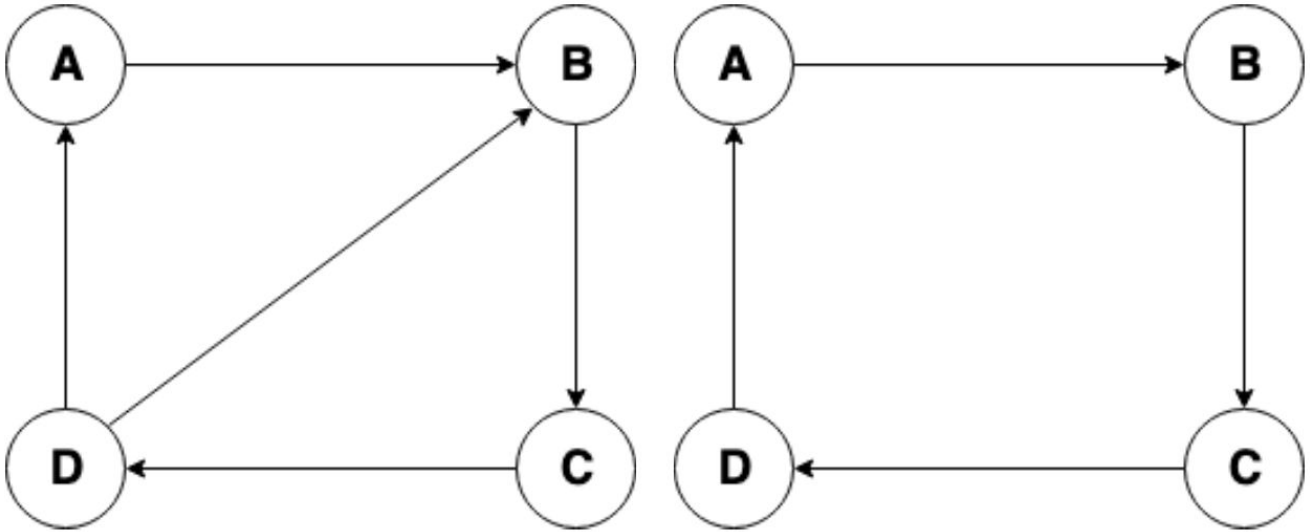
- But we have ***k*** independent hash functions and we only let the element ***x*** through **if all *k*** hash element ***x*** to a bucket of value **1**

m -> number of movies
targets -> number of keys in hasht able (buckets)
k -> number of hash functions

- So, **false positive probability** = $(1 - e^{-km/n})^k$

12 PageRank [10 points]

Figure 3 gives two directed graphs G1=(V, E1), G2=(V, E2) where:
V = $\{A, B, C, D\}$
E1 = $\{(A, B), (B, C), (C, D), (D, A), (D, B)\}$
E2 = $\{(A, B), (B, C), (C, D), (D, A)\}$
Based on Figure 3, answer the following questions:



(a) Directed Graph 1

(b) Directed Graph 2

★ Solution ★

(a) highest: node B, lowest: node A

(b) increase: node A (by reasoning); decrease: node B, C, D

(c) increase: node A; decrease: node B,C,D since page rank is 0.25 for all nodes in this case

(d) page rank is 0.25 for all nodes

(e) increase: node A, B; decrease: node C,D. Straightforward because teleport set is A,B only

Flajolet-Martin

Suppose you are using the Flajolet-Martin algorithm to count the number of distinct elements using the hash function $h_a(x) = (x + a) \bmod 128$
Let the stream you saw so far be 1,3,1,1,6,3,6,1,1

(b) What is the estimated number of distinct elements on using the hash function h_1?

$h_1(1) = 1{+}1 \bmod 128 = 10 \Rightarrow r(1) = 1$
$h_1(3) = 3{+}1 \bmod 128 = 100 \Rightarrow r(3) = 2$
$h_1(6) = 6{+}1 \bmod 128 = 111 \Rightarrow r(1) = 0$
So estimated number of distinct elements is $2^2 = 4$

(c) What is the estimated number of distinct elements on using the hash function h_2?

$h_2(1) = 1{+}2 \bmod 128 = 11 \Rightarrow r(1) = 0$
$h_2(3) = 3{+}2 \bmod 128 = 101 \Rightarrow r(3) = 0$
$h_2(6) = 6{+}2 \bmod 128 = 1000 \Rightarrow r(1) = 3$
So estimated number of distinct elements is $2^3 = 8$

Flajolet-Martin Approach

- Pick a hash function ***h*** that maps each of the ***N*** elements to at least $\log_2 N$ bits
- For each stream element ***a***, let ***r(a)*** be the number of trailing **0s** in ***h(a)***
  - **r(a)** = position of first 1 counting from the right
    - E.g., say ***h(a) = 12***, then ***12*** is ***1100*** in binary, so ***r(a) = 2***
- Record ***R* = the maximum *r(a)* seen**
  - **R = max$_a$ r(a)**, over all the items ***a*** seen so far
- **Estimated number of distinct elements = 2^R**

(d) Give a value of a such that using h_a gives the smallest possible estimate of the number of distinct elements for this stream. Explain in a few words why this is the least possible estimate.

★ **Solution** ★

Since the stream has both odd and even elements, h_a will have atleast one even number for any a. So the estimate of distinct elements is atleast $2^1 = 2$. It suffices to find an a such that its estimate is 2 :
Using any multiple of 4 as a gives this estimate

14 Community Detection [10 points]

Consider the undirected weighted graph below. You are using the Louvain algorithm to perform community detection. Suppose in a certain iteration, after considering nodes 1, 2, and 3, you end up with the following configuration where nodes {1, 2, 3} form a community and nodes {4, 5, 6, 7, 8, 9} form another community.

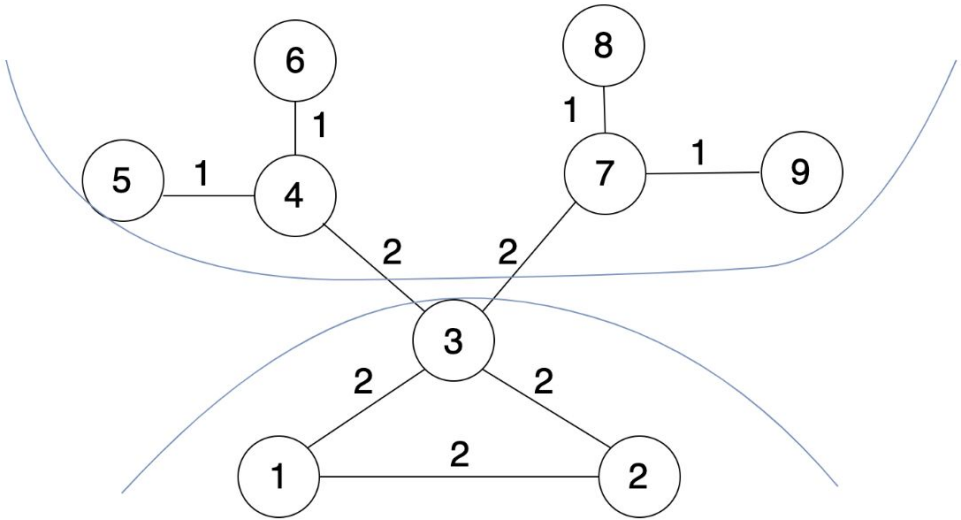


Figure 4: Current Configuration

(a) No

Node 4's contribution if it stays at current cluster:

$$\begin{aligned} Q_{current} &= \frac{1}{2m} * [2 * (1 - \frac{4}{2m}) + 2 * (-\frac{4}{2m}) + (\frac{16}{2m})] \\ &= \frac{1}{2m} * (2 - \frac{32}{2m}) \end{aligned} \tag{3}$$

Node 4's contribution if it joins node 3's cluster:

$$\begin{aligned} Q_{new} &= \frac{1}{2m} * [(2 - \frac{32}{2m}) + 2 * (-\frac{16}{2m})] \\ &= \frac{1}{2m} * (2 - \frac{64}{2m}) \end{aligned} \tag{4}$$

(b) Nothing will change

(c) No, since the top community has 2 disconnected clusters. Idea: after local movement phase, add in a refinement phase where disconnected or poorly connected communities are penalized and refined.

Modularity: 2 Defs

$$Q(G,S) = \frac{1}{2m}\sum_{s\in S}\sum_{i\in s}\sum_{j\in s}\left(A_{ij} - \frac{k_i k_j}{2m}\right)$$

Lecture 11, page 53

Equivalently modularity can be written as:

$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j)$$

- A_{ij} represents the edge weight between nodes i and j;
- k_i and k_j are the sum of the weights of the edges attached to nodes i and j, respectively;
- $2m$ is the sum of all of the edge weights in the graph;
- c_i and c_j are the communities of the nodes; and
- δ is an indicator function

Idea: We can identify communities by maximizing modularity

$$Q(C) \equiv \frac{1}{2m}\sum_{i,j\in C}\left[A_{ij} - \frac{k_i k_j}{2m}\right] = \frac{\Sigma_{i,j\in C} A_{ij}}{2m} - \frac{(\Sigma_{i\in C} k_i)(\Sigma_{j\in C} k_j)}{(2m)^2}$$

$$= \frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2$$

Links within the community

Total links

Louvain: Modularity Gain

- **What is ΔQ if we move node i to community C?**

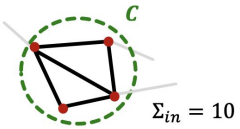
$$\Delta Q(i \to C) = \left[\frac{\sum_{in}+k_{i,in}}{2m} - \left(\frac{\sum_{tot}+k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$$

- Also need to derive $\Delta Q(D \to i)$ of taking node i out of community D.

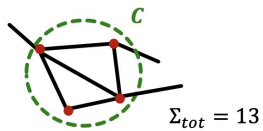
- And then: $\Delta Q = \Delta Q(i \to C) + \Delta Q(D \to i)$

- **Define:**
  - $\boldsymbol{\Sigma_{in}} \equiv \sum_{i,j \in C} A_{ij}$... sum of link weights between nodes in C
  - $\boldsymbol{\Sigma_{tot}} \equiv \sum_{i \in C} k_i$... sum of all link weights of nodes in C

$\boldsymbol{\Sigma_{in}}$:

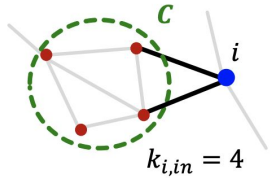


$\boldsymbol{\Sigma_{tot}}$:

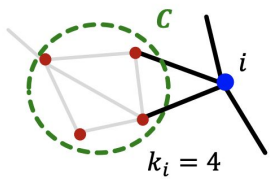


- **Further define:**
  - $\boldsymbol{k_{i,in}} \equiv \sum_{j \in C} A_{ij} + \sum_{j \in C} A_{ji}$ ... sum of link weights connecting node i and C
    - **(note that each edge gets counted twice, see formula)**
  - $\boldsymbol{k_i}$... sum of all link weights (i.e., degree) of node i

$\boldsymbol{k_{i,in}}$:



$\boldsymbol{k_i}$:



(c) Learning BFS with GNN [7pts]

Next, we investigate the expressive power of GNN for learning simple graph algorithms. Consider breadth-first search (BFS), where at every step, nodes that are connected to already visited nodes become visited. Suppose that we use GNN to learn to execute the BFS algorithm. Suppose that the embeddings are 1-dimensional. Initially, all nodes have input feature 0, except a source node which has input feature 1. At every step, nodes reached by BFS have embedding 1, and nodes not reached by BFS have embedding 0. 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.

★ **SOLUTION:** The message function: identity mapping, $M(h_v^k) = h_v^k$.

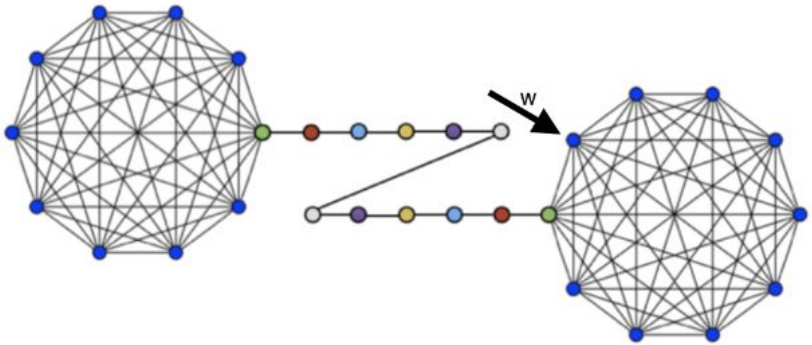
The aggregation function: should output 1 if at least one of the neighboring nodes have feature 1, and 0 otherwise. Example answer: $h_{N(v)}^{k+1} = sgn(\sum_{u \in N(v)} M(h_u^k))$ (using clamp or max is also fine)

The update function: should output 1 if either the feature of the node in the previous layer or the aggregation of its neighbors' features is 1 , and 0 otherwise. Example answer: $h_v^{k+1} = sgn(h_v^k + h_{N(v)}^{k+1})$(using clamp or max is also fine)

Suppose f = the structural distance between u and v when considering their k-hop neighborhoods

Edge weight w(u,v) = e^(-f)

If the structural distance is small, then the edge weight is large, therefore corresponding to the structural similarity



P15, Solution:

A: The vector representations will form two groups(clusters). In each cluster, the node representations are very similar to each other.(Full credit for mentioning two clusters.) No, this is not desirable since the nodes on the two balls are exactly the same structurally.

B: node2vec:All the nodes on the right clique.(Also correct if the student answers all the neighbor nodes)
struct2vec:Any node in G

C: For broader or narrower context and better representations of the structures.(Full credit for reasonable answers.)

D: The vector representations will form only one group(cluster). The node representations are very similar to each other.(Full credit for mentioning they have similar embeddings.)

16 Locality Sensitive Hashing [15 points]

16.1 LSH Application: Finding Similar Documents

In this question, you will apply Locality-Sensitive Hashing to efficiently find candidate document pairs that likely have high similarity.

(a) Assume to use single words as tokens, and to convert documents into sets of 2-shingles. Recall that you can use Jaccard similarity as a similarity measure for shingled documents. Consider the following two documents (pre-processed to remove punctuation and converting all letters to lower-case):

D_1 = the quick brown fox jumps over the lazy dog

D_2 = jeff typed the quick brown dog jumps over the lazy fox by mistake

The Jaccard similarity of D_1 and D_2 on 2-shingles is: $sim(D_1, D_2) =$_____.

On 2-shingles, $|D_1 \cup D_2| = 15$, $|D_1 \cap D_2| = 5$.
Therefore, the Jaccard similarity is $sim(D_1, D_2) = \frac{|D_1 \cap D_2|}{|D_1 \cup D_2|} = \frac{1}{3}$.
(A common mistake is counting using 1-shingles instead of 2-shingles. Another common mistake is calculating the Jaccard **distance** instead, which equals $1 - sim(D_1, D_2) = \frac{2}{3}$.)

$$\pi = \begin{bmatrix} 3 & 1 \\ 6 & 3 \\ 1 & 5 \\ 2 & 6 \\ 4 & 2 \\ 5 & 4 \end{bmatrix} \qquad A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad M = \begin{bmatrix} -- & -- & -- & -- \\ -- & -- & -- & -- \end{bmatrix}$$

$$M = \begin{bmatrix} 3 & 2 & 1 & 2 \\ 1 & 3 & 4 & 6 \end{bmatrix}$$

b bands, r rows/band

- Say columns C_1 and C_2 have similarity t
- Pick any band (r rows)
  - Prob. that all rows in band equal = t^r
  - Prob. that some row in band unequal = $1 - t^r$
- Prob. that no band identical = $(1 - t^r)^b$

False negative if sim > s

- Prob. that at least 1 band identical = $1 - (1 - t^r)^b$

False positive if sim < s

(c) $1-(1-0.5^5)^{10} \approx 0.272$, $1-(1-0.8^5)^{10} \approx 0.981$.

Assume to use the LSH algorithm with $b = 10$ bands and $r = 5$ rows per band.
If a pair of documents D_3, D_4 have Jaccard similarity $sim(D_3, D_4) = 0.5$, then the probability that they have identical hash values in **at least** 1 band is: _____.
If a pair of documents D_5, D_6 have Jaccard similarity $sim(D_5, D_6) = 0.8$, then the probability that they have identical hash values in **at least** 1 band is: _____.
(You can leave exponents in your expression without calculating them.)

(d) Assume Emily and Pierre are using LSH with inputs from M different Min-hash functions. If Emily cares more about higher precision (i.e., less false positives) while Pierre cares more about higher recall (i.e., less false negatives), and their target similarity threshold s are the same, who should divide the M Min-Hash functions into **more** bands (by setting the number of bands b higher, with each band containing fewer rows)?
Circle exactly one option below.

A. Emily, who cares more about higher precision, should set the number of bands b higher

B. Pierre, who cares more about higher recall, should set the number of bands b higher

C. They should choose the same number of bands

D. Changing the number of bands b does not affect precision and recall

Precision = TP/(TP+FP)

Recall = TP/(TP+FN)

(e) Given a fixed number M of input Min-Hash functions, what will happen if the number of bands b is set too high, and each band contains very few rows?

(d) B, because for fixed $M = b * r$ and s, setting b higher leads to less false negatives, although it will lead to more false positives at the same time.

(e) When the number of bands b is set too high, intuitively it is very easy for two document pairs to have identical hash values in at least 1 band.
Therefore, LSH will return a lot of candidate pairs, many of which are not actually similar to each other. In other words, we will get many false positives, but very few false negatives (low precision, high recall), meaning that LSH filters candidate pairs less aggressively.

16.2 Theory of LSH

(f) For a given hash family H, we have $\forall h \in H, h(x = y) = Similarity(x, y)$. Suppose you construct a $(3, 4, 5)$ way OR-AND-OR hash family G from H. Given similarity s, what is the probability of two candidate pairs get hashed into the same bucket for each $g \in G$?

$$1 - (1 - (1 - (1 - s)^3)^4)^5$$

(g) How many hash function from H do you need to construct a hash family composed of $(3, 4, 5)$ OR-AND-OR construction followed by $(6, 7)$ AND-OR construction?

3*4*5*6*7

- Take points ***x*** and ***y*** s.t. ***Pr[h(x) = h(y)] = s***
  - ***H*** will make **(x,y)** a candidate pair with prob. **s**
- Construction makes **(x,y)** a candidate pair with probability ***1-(1-s^r)^b*** **The S-Curve!**
  - **Example:** Take **H** and construct **H'** by the **AND** construction with ***r* = 4**. Then, from **H'**, construct **H''** by the **OR** construction with ***b* = 4**